

“Resumen” de Teoría de las Comunicaciones

Tomás C.

Septiembre-Octubre-Noviembre, 2021



Argentinosaurus huinculensis, E.D. Rodríguez.

Acerca de este resumen:

Este es un compilado con los conceptos teóricos de la materia “Teoría de las Comunicaciones” (a.k.a. *Redes*), escrito en base a las clases teóricas y prácticas del primer cuatrimestre de 2020 (cursada a distancia), y a apuntes hechos por mí en ese momento. También incorporé contenido de distintos resúmenes escritos por otros estudiantes, y de la última clase teórica de la cursada del segundo cuatrimestre de 2021, dado que ésta no fue dada en la cursada mencionada antes.

La parte teórica de la materia en el primer cuatrimestre de 2020 estuvo a cargo de Rodrigo Castro (profesor), y la parte práctica de Marcos Cervetto y Esteban Lanzarotti (jefes de trabajos prácticos), mientras que los ayudantes de la cursada fueron Leonardo Balbiani, Paula Verghelet, Martín Medina y Julián Len.

Por más de que me limité básicamente a incluir lo que escribieron y dijeron los docentes en sus clases, es posible que el “resumen” (las comillas son porque quedó muy largo, aunque realmente considero que los contenidos están resumidos y no transcritos) contenga algún error conceptual (además de errores ortográficos o de redacción).

Índice

1	Introducción	1
1.1	Conmutación de Circuitos	1
1.2	Conmutación de Paquetes	1
1.3	Arquitectura de Redes	1
1.4	Fuentes	3
2	Teoría de la Información	4
2.1	Introducción	4
2.1.1	Fuente de Memoria Nula	5
2.1.2	Entropía	5
2.2	Codificación	6
2.2.1	Código Bloque	6
2.2.2	Código no Singular	6
2.2.3	Código Unívocamente Decodificable	6
2.2.4	Código Instantáneo	6
2.2.5	Longitud de Código	7
2.2.6	Codificador Óptimo	7
2.3	Problemas en Medios de Transmisión Reales	7
2.4	Teorema de Capacidad del Canal de Shannon	9
2.5	Fuentes	9
3	Nivel Físico	10
3.1	Fundamentos	10
3.2	Señales	10
3.2.1	Ondas	11
3.3	Ancho de Banda	11
3.4	Medios de Transmisión	12
3.5	Red Telefónica	12
3.5.1	Estructura del Sistema Telefónico	12
3.5.2	Multiplexación	13
3.6	Teorema de Muestreo	14
3.7	Conversión Analógico-Digital	14
3.8	Modulación	15
3.8.1	Moduladora Analógica y Portadora Analógica	15
3.8.2	Moduladora Digital y Portadora Analógica	15
3.8.3	Moduladora Analógica y Portadora Digital	16
3.8.4	Moduladora Digital y Portadora Digital	16
3.8.5	Capacidad de Canal y Modulación	17
3.8.6	Bit Error Rate (BER) y Modulación	17
3.9	Redes Inalámbricas	17
3.10	Fuentes	18
4	Nivel de Enlace: Protocolos Punto a Punto	19
4.1	Fundamentos	19
4.2	Encapsulamiento (framing)	19
4.2.1	Eficiencia de un Frame	19
4.3	Tipos de Servicio	19
4.4	Detección y Corrección de Errores	20
4.5	Transmisión Confiable	20

4.5.1	Stop & Wait	20
4.5.2	Eficiencia de un Protocolo	21
4.5.3	Sliding Window	22
4.6	Delay, Propagación y Transmisión	23
4.7	Capacidad de Volumen	24
4.8	Fuentes	24
5	Nivel de Enlace: Protocolos de Acceso Múltiple	25
5.1	Acceso a Medios Compartidos	25
5.1.1	Problema de Acceso	25
5.2	Ethernet (IEEE 802.3)	25
5.3	Mecanismo de Acceso: CSMA/CD	26
5.3.1	Colisiones	27
5.3.2	Retransmisiones	28
5.3.3	Performance	28
5.3.4	Ventajas y Desventajas	29
5.4	Logical Link Control	30
5.5	Local Area Network	30
5.5.1	Topologías de Red y Dominios	30
5.5.2	LAN Extendida con 802.2	31
5.6	Learning Bridges	32
5.6.1	El Problema de las Topologías con Ciclos	33
5.7	Spanning Tree Protocol	34
5.7.1	Idea	34
5.7.2	Mecanismo	35
5.7.3	Bridge Protocol Data Units	35
5.7.4	Estados de Interfaces	35
5.8	LAN Virtual	36
5.9	Problemas en Redes Inalámbricas	36
5.9.1	Problema de la Estación Oculta	36
5.9.2	Problema de la Estación Expuesta	36
5.10	CSMA/CA	37
5.11	Wi-Fi (IEEE 802.11)	37
5.11.1	Anomalía del Wi-Fi	39
5.11.2	MACA y MACAW	40
5.12	Fuentes	40
6	Nivel de Red: Ruteo	41
6.1	Introducción	41
6.1.1	Sistemas Autónomos	41
6.1.2	Ruteo Interno y Externo	41
6.1.3	Forwarding versus Routing	41
6.1.4	Ruteo Estático y Dinámico	42
6.2	Protocolos de Ruteo Interno	42
6.2.1	Distance-vector	42
6.2.2	Link-state	44
6.2.3	Distance-vector vs Link-state	46
6.3	Protocolos de Ruteo Externo	47
6.3.1	Border Gateway Protocol (BGP)	48
6.3.2	BGP y los Network Access Points	48

6.4	Fuentes	48
7	Nivel de Red: IP	49
7.1	Internetworking	49
7.2	Conmutación de Paquetes	49
7.3	Conmutación sin Conexión (datagramas)	50
7.4	Conmutación Orientada a Conexión (circuitos virtuales)	50
7.4.1	Tipos de Conexiones	51
7.5	Datagrama vs Circuito Virtual	52
7.6	Internet Protocol (IP)	52
7.6.1	Modelo de Servicio IP	52
7.6.2	El Encabezado IPv4	53
7.6.3	Fragmentación	53
7.6.4	Direccionamiento Global	54
7.6.5	Direcciones IP Classful	54
7.6.6	Direcciones y Máscaras	55
7.6.7	Direcciones Especiales	56
7.6.8	Asignación de Direcciones IP	57
7.6.9	Direcciones IP Privadas	57
7.6.10	Forwarding	57
7.6.11	Subredes IP	58
7.6.12	Máscaras de Tamaño Variable (VLSM)	59
7.6.13	Direcciones sin Clases (CIDR)	59
7.7	ARP	59
7.8	ICMP	60
7.9	Fuentes	61
8	Nivel de Transporte	62
8.1	Introducción	62
8.2	Enlace de Datos versus Transporte	62
8.3	Transmission Control Protocol (TCP)	63
8.3.1	Conceptos Generales	63
8.3.2	Maximum Segment Size	64
8.3.3	Segmento TCP	64
8.3.4	Establecimiento de la Conexión	66
8.3.5	Finalización de la Conexión	67
8.3.6	Ventana Deslizante	67
8.3.7	Control de Flujo	69
8.3.8	Máquina de Estados Finitos	69
8.3.9	Retransmisión y Time-out	70
8.4	User Datagram Protocol (UDP)	71
8.4.1	Conceptos generales	71
8.4.2	Multiplexación mediante Puertos	72
8.4.3	Segmento UDP	72
8.4.4	Modelo de Servicio	73
8.5	Congestión	73
8.5.1	Administración de Buffers	73
8.5.2	Definición y Soluciones	74
8.5.3	Análisis de Congestión	74
8.5.4	Métricas de Detección	75

8.5.5	Causas de la Congestión	76
8.5.6	Control de Congestión	76
8.5.7	Criterios de Evaluación	77
8.5.8	Congestión y Calidad de Servicio	78
8.5.9	Teoría de Control	78
8.5.10	Random Early Detection (RED)	79
8.5.11	Flow Random Early Detection (FRED)	81
8.5.12	Control de Congestión en TCP	81
8.5.13	Performance de TCP	84
8.6	Fuentes	85
9	Nivel de Aplicación	86
9.1	Introducción	86
9.2	Paradigma tipo Cliente-Servidor	86
9.3	Servicios de Transporte	87
9.3.1	Criterios de Selección	88
9.4	Domain Name System (DNS)	88
9.4.1	Fundamentos	88
9.4.2	Funcionamiento General	89
9.4.3	El Espacio de Nombres	89
9.4.4	Registros DNS	89
9.4.5	Consultas DNS	90
9.4.6	Respuestas DNS	90
9.4.7	Proceso de una Consulta DNS	91
9.5	Correo Electrónico	92
9.5.1	Componentes Principales	92
9.5.2	Simple Mail Transfer Protocol (SMTP)	92
9.5.3	Multipurpose Internet Mail Extensions (MIME)	94
9.5.4	Transferencia y Entrega de Mensajes	94
9.6	Web y HyperText Transfer Protocol (HTTP)	96
9.6.1	Fundamentos	96
9.6.2	Operación	96
9.6.3	Mensajes HTTP	97
9.6.4	Tiempo de Respuesta	99
9.6.5	Conexiones HTTP	99
9.6.6	Cookies: Mantenimiento del Estado	100
9.7	Peer-to-Peer y BitTorrent	101
9.8	Fuentes	101
10	Seguridad en Redes	102
10.1	Conceptos	102
10.2	Protocolos y Capas	102
10.3	Introducción a la Criptografía	103
10.3.1	Conceptos	103
10.3.2	Métodos Básicos de Cifrado	103
10.3.3	Principio de Kerckhoff	104
10.3.4	One-Time Pads	104
10.3.5	Cifrado de Bloque Iterativo	104
10.3.6	Data Encryption Standard (DES)	105
10.4	Criptografía de Clave Simétrica	105

10.5	Criptografía de Clave Asimétrica	106
10.5.1	Algoritmo de Rivest-Shamir-Adleman (RSA)	107
10.6	Firma Digital	108
10.6.1	Message Digest	108
10.7	Autenticación	109
10.7.1	Autenticación de Dos Vías	110
10.7.2	Ataque por Sesiones	110
10.7.3	Distribución de Claves Simétricas Confiable	111
10.7.4	Certificados Digitales	112
10.8	Secure Sockets Layer (SSL) y Transport Layer Security (TLS)	112
10.8.1	Handshake SSL	113
10.9	Firewall	113
10.10	Ataques de Red	114
10.10.1	Sniffing	114
10.10.2	Spoofing	114
10.10.3	Hijacking	114
10.10.4	Ingeniería Social	114
10.10.5	Explotar Bugs	114
10.10.6	Confianza Transitiva	114
10.10.7	Ataques Dirigidos por Datos	115
10.10.8	Caballo de Troya	115
10.10.9	Denegación de Servicio (DoS)	115
10.10.10	Enrutamiento Fuente	115
10.10.11	Adivinación de Contraseñas	115
10.10.12	Mensajes de Control de Red	115
10.11	Fuentes	116

1. Introducción

1.1. Conmutación de Circuitos

El telégrafo fue el antecesor del teléfono, un primer acercamiento a la comunicación de mensajes vía una codificación. Desde fines de siglo XIX hasta segunda mitad del siglo XX, aparecen las centrales de conmutación de circuitos (centrales telefónicas). A estas centrales llegaban señales (cables) correspondientes a todas las casas que participaban en el sistema de teléfonos. Las operadoras conectaban dos circuitos en sus tableros para cerrar el circuito y permitir la comunicación entre las dos partes involucradas.

1.2. Conmutación de Paquetes

Ideas evolucionando desde fines de los 50 a fines de los 60, la *semilla* de Internet.

Objetivo: tener una red **más tolerante a fallas** (si una central de conmutación de circuitos dejaba de estar disponible por algún motivo de fuerza, todas las personas pertenecientes a esa zona se verían incomunicadas), y **más flexible** a la hora de conectar dos puntos distantes. Además, se quería una red que **escale más fácilmente** ante un incremento en el acceso a la comunicación.

Estrategia: una red descentralizada con múltiples caminos entre dos puntos, y dividir los mensajes en fragmentos, que podrían seguir caminos diferentes (habrá que tener cuidado con el orden en el que llegan...).

Ejemplos: DARPA, RAND, UCLA, MIT, NPL.

Hoy en día las tecnologías relacionadas con Internet evolucionan principalmente impulsadas por las aplicaciones que usa todo el mundo a diario (compras, entretenimiento, mensajería, juegos, etc.).

Cómo entender técnicamente la infraestructura involucrada en todo esto: con las arquitecturas de red y los protocolos de red; son estándares que se siguen para que pueda coexistir la gran variedad de aplicaciones que son utilizadas cada minuto, en una misma red.

1.3. Arquitectura de Redes

La figura 1 nos muestra a la izquierda un Host A (que podría ser una computadora, un server, etc.), y a la derecha un Host B. Estos hosts tienen funcionando dentro de sí un “sandwich” de algoritmos, con funciones bien delimitadas y definidas para cada uno. Las capas se comunican entre sí, pero tienen responsabilidades divididas. De la capa superior (aplicación, podría ser un navegador de internet), se pasa por todas las capas subyacentes hasta llegar a la capa física, donde la señal física pasa por una *nube* hasta llegar al otro host, que tiene también las mismas capas, en este caso usadas para decodificar lo que recibió.

A mediados de los 80 había muchos tipos de redes, no unificadas; en 1983 aparece una publicación de ISO para establecer un acuerdo, un estándar en la comunicación. Fue una propuesta de una **arquitectura única de red**, que uniformice la forma de construir las redes de comunicación: el modelo OSI-ISO (Open Systems Interconnection).

La idea es que las capas tienen **dos tipos de capacidades de comunicación**: entre capas de un mismo host (layer-to-layer communication) y entre la misma capa de dos hosts (peer-layer communication).

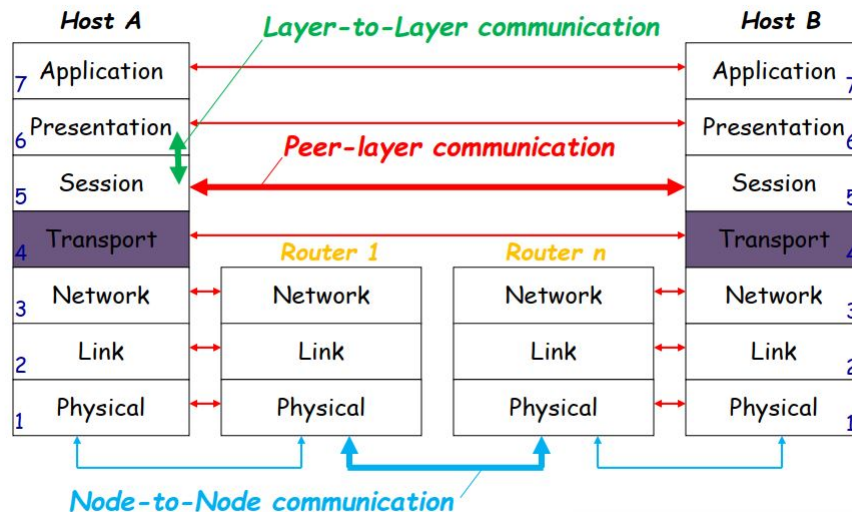


Figura 1: Modelo OSI-ISO

Las capas tienen tareas definidas, y no saben cómo realizar otras tareas, así que se ocupan de **delegar** estas responsabilidades en otras capas: se piden cosas entre capas contiguas. Cada capa sabe enviar información a la capa de arriba, y pedir información a la capa de abajo.

Modelo OSI			
	Unidad de dato	Capa	Función
Host	Datos	7. Aplicación	Servicio de red para aplicación
		6. Presentación	Representación de datos, cifrado, estandarización.
		5. Sesión	Comunicación entre hosts, manejo de sesiones entre aplicaciones.
	Segmento	4. Transporte	Ruteo confiable de paquetes entre nodos de la red.
Medios	Paquete / Datagrama	3. Red	Direccionamiento, ruteo no confiable de datagramas entre nodos.
	Bit / Frame	2. Enlace	Conexión confiable en enlace punto a punto.
	Bit	1. Físico	Conexión no confiable en enlace punto a punto.

Cada capa habla un lenguaje propio, entiende conceptos y métricas propias, y asume que en el otro extremo de la comunicación, el nodo receptor tiene una capa similar a la suya, que va a poder entender lo enviado.

En cuanto a la información por nivel, cada capa implica el agregado de **información de control** en forma de **encabezados** (*headers*). Un emisor quiere tener ciertas garantías acerca de la recepción del tren de bits que envió, por lo que en cada capa inferior se le van agregando estos encabezados al mensaje original enviado desde la capa de aplicación.

El objetivo de una capa consiste en proveer un **servicio a la capa superior**:

- Confiabilidad: ¿es confiable o no?

- Control de errores: ¿se produjo algún error? ¿qué se hace con él?
- Control de flujo: ver nivel de transporte

Para esto, la estrategia es el encapsulamiento o **framing**: se encapsulan los bits del mensaje en *frames*, agregando información de control.

Las capas inferiores (red, enlace y física) tienen una particularidad: el nodo *peer* que esperan que interprete la información enviada por ellas no es el nodo en el otro extremo (el destinatario final), sino que saben que pueden estar hablando con algún nodo intermedio, que pasará el mensaje en dirección al nodo final (se llama **node-to-node**). Estos nodos intermedios normalmente son elementos de red (switches, hubs, routers, etc.). La siguiente capa hacia arriba (transporte) es la primera que va desde un extremo hasta el otro (se llama **end-to-end**).

Lo que se aplica en la realidad no es el modelo OSI tal cual, sino una simplificación: el modelo TCP/IP, que colapsa algunas capas (tiene cuatro en vez de siete), como se puede ver en la figura 2

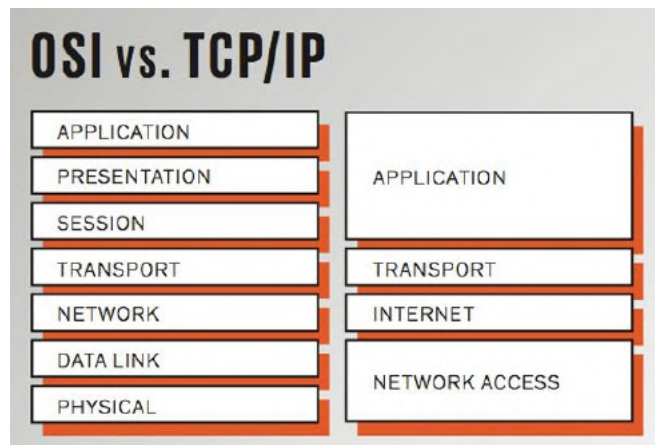


Figura 2: A la izquierda, las capas del modelo OSI, y a la derecha las del modelo TCP/IP, que no posee los niveles de sesión y presentación.

1.4. Fuentes

- Rodrigo Castro. Teoría de las Comunicaciones, Clase Teórica 1. Primer cuatrimestre, 2020.
- Julián Sackmann. Teóricas de Teoría de las Comunicaciones. 2012.

2. Teoría de la Información

2.1. Introducción

C. Shannon, 1948: *A mathematical theory of communication*:

- Establece las bases para las comunicaciones digitales.
- Propone un modelo para pensar **cualquier tipo de comunicación** (radio, televisión, teléfono...).
- Cualquier mensaje, sin importar el canal, está en riesgo de una entrega incorrecta por culpa del **ruido**.
- La clave para superar el ruido y asegurar una entrega confiable de mensajes, es la información contenida en el mensaje: **cuánta información** contiene el mismo.
- El significado (semántica) del mensaje es irrelevante para su transmisión: concibe a los mensajes como secuencias con propiedades estadísticas.
- Cuanto mayor es la **entropía** (grado de *sorpres*a con el cual una fuente emite símbolos, cantidad de información no redundante, no predecible) del mensaje, más esfuerzo es necesario para transmitirlo.

La teoría de Shannon se basa en dos teoremas fundacionales: codificación para una **fente sin ruido**, y codificación para un **canal ruidoso**.

Uno de ellos describe la máxima eficiencia posible de un método de corrección de errores (codificación) frente a los niveles de ruido y corrupción de los datos.

- No dice nada sobre cómo implementar esa codificación, pero da un **límite teórico absoluto para la transmisión de bits**.

Información: sea E un suceso que puede presentarse con probabilidad $P(E)$. Cuando E tiene lugar, se dice que se han recibido

$$I(E) = \log \frac{1}{P(E)}$$

unidades de información.

- Es la inversa de la probabilidad, así que si un símbolo tiene mucha probabilidad de ser emitido por una fuente, entonces la información aportada por él es muy poca. Contrariamente, si la probabilidad de que aparezca es baja, la información recibida será alta.
- Si la base del logaritmo se toma como 2, la unidad correspondiente a la cantidad de información es el *bit*.
- Si $P(E) = 0.5$, ocurre que $I(E) = 1$ bit. Por lo tanto, *un bit es la cantidad de información obtenida al especificar una de dos posibles alternativas igualmente probables*.

2.1.1. Fuente de Memoria Nula

Es una fuente en la que los símbolos emitidos son estadísticamente independientes. Es decir, se puede describir completamente con el alfabeto fuente y las probabilidades con la que sus símbolos se presentan.

La probabilidad de que se emita un símbolo depende de un proceso puramente probabilístico, y no de los símbolos emitidos en el pasado.

2.1.2. Entropía

Si $P(s_i)$ es la probabilidad de que aparezca el símbolo s_i , entonces la cantidad media (esperanza) de información por símbolo para una fuente S es:

$$\sum_{s_i \in S} P(s_i) I(s_i)$$

Esta magnitud se denomina **entropía** $H(S)$ de una fente de memoria nula S :

$$\begin{aligned} H(S) &= \sum_{s_i \in S} P(s_i) \log_2 \frac{1}{P(s_i)} \quad \text{bits} \\ &= - \sum_{s_i \in S} P(s_i) \log_2 P(s_i) \quad \text{bits} \end{aligned}$$

Interpretaciones de la entropía:

- El valor medio ponderado de la cantidad de información del conjunto de mensajes posibles.
- La medida de la incertidumbre promedio (grado de incerteza) acerca de una variable aleatoria (que sería el próximo símbolo que va a aparecer).
- La cantidad de información obtenida al observar la aparición de cada nuevo símbolo.

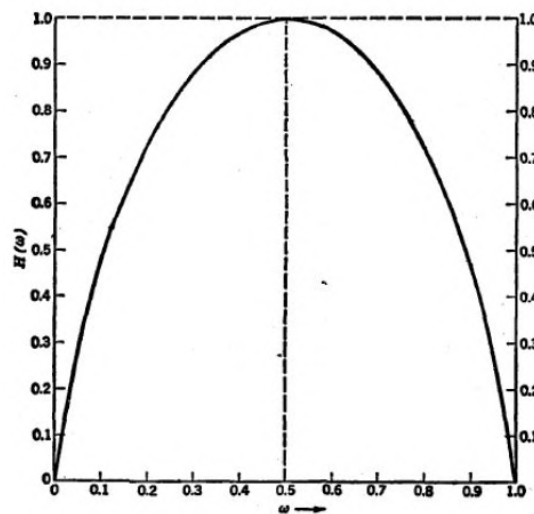


Figura 3: Función de entropía

Propiedades de la entropía:

- Es no negativa, y se anula si y sólo si un estado de la variable es igual a 1 y el resto es 0.
- Se maximiza cuando todos los símbolos tienen la misma probabilidad de aparición, y se minimiza cuando uno tiene probabilidad 1 y el otro 0, como se ve en la figura 3. Si hay n estados equiprobables, entonces $P(s_i) = 1/n \forall i \in \{1, \dots, n\}$, con lo cual $H(S) = \log_2 n$.

2.2. Codificación

Alfabeto: conjunto de símbolos.

Código: dado un alfabeto fuente Σ , un *código* es una correspondencia entre todas las secuencias posibles de símbolos de Σ a secuencias de símbolos de un alfabeto de código X .

El objetivo es lograr una **representación eficiente de la información**, a través de eliminar redundancia.

2.2.1. Código Bloque

- Es un código $C : \Sigma \rightarrow X^*$ que asigna a cada símbolo de Σ una secuencia fija de símbolos de X .
- Ejemplo: $C_1 : \{s_1, s_2, s_3, s_4\} \rightarrow \{0, 1\}^*$

$$C_1(s_1) = 0$$

$$C_1(s_2) = 11$$

$$C_1(s_3) = 01$$

$$C_1(s_4) = 101$$

2.2.2. Código no Singular

- Un código es *no singular* si todas sus palabras son distintas.
- Equivalentemente, si C es una función inyectiva.
- Ejemplo: el C_1 de arriba es un código no singular

2.2.3. Código Unívocamente Decodificable

- Un código es *unívocamente decodificable* si ninguna tira de símbolos del alfabeto de código admite más de una única decodificación.
- Formalmente: si su extensión de orden n es no singular para todo n natural.

2.2.4. Código Instantáneo

- Un código es *instantáneo* cuando es posible decodificar las palabras sin necesidad de conocer los símbolos que la suceden.

- La condición *necesaria y suficiente* para que un código sea instantáneo es que sus palabras cumplan la condición de los prefijos: que no exista una palabra que sea prefijo de otra palabra de longitud mayor. La consecuencia de esto es que apenas el receptor recibe una palabra, está seguro de cuál es el símbolo que recibió, sin tener que esperar a que lleguen más bits.
- **Teorema:** si un código es instantáneo, entonces es unívocamente decodificable

2.2.5. Longitud de Código

Dado un código C sobre una fuente S , la *longitud media* de C se define como:

$$L(C) = \sum_{s \in S} |C(s)| P_S(s)$$

donde $|C(s)|$ es la longitud de la codificación del símbolo s .

2.2.6. Codificador Óptimo

- Un codificador óptimo es aquel que usa el **menor número posible de bits** para codificar un mensaje. Es decir, un código se dice óptimo si no existe un código para la misma fuente con menor longitud media.
- La expresión $\lceil \log(1/P(x)) \rceil$ representa el número de bits necesario para codificar el mensaje x en un codificador óptimo.
- **Eficiencia** de un código: $H(S)/L(C)$
- Para tener un código eficiente, la idea es asignar las palabras más cortas a los símbolos más probables.
- La **codificación de Huffman** es un método que obtiene codificadores óptimos, armando un árbol de prioridades en base a la frecuencia de aparición de cada símbolo en un mensaje.
- **Teorema** (codificación sin pérdida de información): $H(S) \leq L(C)$
- Si un código satisface el teorema anterior, se dice que codifica *sin pérdida de información*.

2.3. Problemas en Medios de Transmisión Reales

Un medio real está sujeto a ruido, y a condiciones no ideales:

- Un mensaje, que sale de una fuente, pasa luego por un codificador para tal fuente, donde se obtiene un mensaje con una cantidad L de bits a partir de una codificación idealmente eficiente (eliminando redundancias).
- Este luego debe pasar por un codificador para el canal, donde muchas veces se le agrega redundancia, pasando a tener una palabra de N bits. Esto se debe a que el canal estará sujeto a ruido, y la redundancia puede ayudar a combatirlo (e.g. si se quiere enviar un 1, se envían 100 1s por si se pierden algunos).

- Del otro lado del canal, ocurre el proceso inverso: un decodificador recibe el código de N bits, y lo decodifica para volver a tener el mensaje de L bits.
- Por último, el decodificador del destinatario se encarga de decodificar el mensaje una última vez, para obtener el mensaje original enviado por la fuente.

Pueden ocurrir perturbaciones en la transmisión:

- La señal recibida puede diferir de la transmitida.
- Esto puede deberse a causas analógicas (una degradación en la calidad de la señal), o digitales (errores de bits causados por atenuación y distorsión de atenuación o de retardo, al ruido, etc.).

Atenuación

- La intensidad de la señal disminuye con la distancia, dependiendo del medio de transmisión.
- Esta intensidad debe ser suficiente para ser detectada, y suficientemente mayor que el ruido para que se reciba sin errores.
- Se ven más afectadas las frecuencias mayores, por lo que se puede utilizar ecualización para amplificarlas.

Distorsión de retardo

- Se da sólo en medios guiados (e.g. cables).
- La velocidad de propagación varía con la frecuencia, por lo que las diferentes componentes frecuenciales llegan al receptor en distintos instantes, lo que produce desplazamientos de fase entre ellas.
- Para una señal limitada en frecuencia, la velocidad es mayor cerca de la frecuencia central.

Ruido

- Son señales adicionales insertadas entre el transmisor y el receptor.
- Ruido térmico: se debe a la agitación térmica de los electrones, aumenta con la temperatura absoluta y está uniformemente distribuido en la frecuencia.
- Ruido por intermodulación: señales que son la suma y la diferencia de las frecuencias originales y sus múltiplos; se produce por falta de linealidad en el canal.
- Ruido por diafonía: se da cuando la señal de una línea interfiere con otra.
- Ruido impulsivo: son impulsos irregulares, que van y vienen (e.g. interferencia electromagnética externa).

Velocidad de transmisión: C , en bits por segundo.

Ancho de banda: B , en Hertz (limitado por el transmisor y el medio).

Ruido: N (nivel medio a través del canal). Para un cierto nivel de ruido, a mayor velocidad C , menor período de un bit, y mayor tasa de error.

Tasa de errores: BER (*bit error rate*, cantidad de veces que se cambia un 1 por 0 o viceversa, en una unidad de tiempo).

Relación señal a ruido: SNR (*signal-noise ratio*), definida como $SNR = \frac{\text{potencia de la señal}}{\text{potencia del ruido}}$, o, en decibels: $SNR_{dB} = 10 \log_{10}(SNR)$. A menor SNR , más significativo es el ruido respecto a la señal a transmitir; y si la SNR es grande, es más factible que la transmisión del mensaje tenga éxito.

2.4. Teorema de Capacidad del Canal de Shannon

Si se aumentan el ancho de banda B y la potencia de señal S , aumenta la velocidad de transmisión C . Sin embargo, un aumento en el ancho de banda B aumenta el ruido N (porque puede haber ruido en más frecuencias), y un aumento en la potencia de señal S aumenta las no linealidades y el ruido de intermodulación.

Es por esto que aumentar arbitrariamente el ancho de banda (o la potencia de la señal) no es una solución para mejorar la velocidad de transmisión.

Según Shannon, la *velocidad binaria teórica máxima de transmisión* (o **capacidad**) para un canal es:

$$C_{max} = B \log_2(1 + SNR)$$

2.5. Fuentes

- Rodrigo Castro. Teoría de las Comunicaciones, Clase Teórica 1. Primer cuatrimestre, 2020.
- Esteban Lanzarotti. Teoría de las Comunicaciones, Clase Práctica 1. Primer cuatrimestre, 2020.
- Norman Abramson, 1963. *Information Theory and Coding*. Capítulos 1 y 2.

3. Nivel Físico

3.1. Fundamentos

El esquema central para estudiar los sistemas de comunicaciones (ver figura 4), propuesto por C. Shannon, tiene los siguientes elementos centrales:

- Una **fuente de información**: emite los símbolos a transmitir, que componen un **mensaje**.
- Ese mensaje es pasado a un **transmisor**, que producirá una **señal**, y usará algún **canal**, un medio físico.
- Ese canal tiene propiedades muy diferentes dependiendo de circunstancias propias del sistema de comunicación, y estará sujeto a **ruido**: no existe un canal ideal. Entonces el objetivo es hacer un uso eficiente del canal.
- La señal es luego recibida por un **receptor**, que finalmente le entrega el mensaje decodificado al **destinatario**.

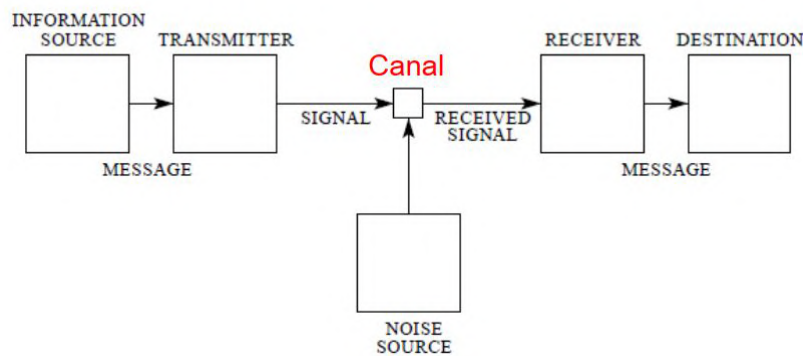


Figura 4: Esquema de un sistema de comunicación

3.2. Señales

Hay dos tipos fundamentales de señales: las analógicas y las digitales.

- Las **señales analógicas** tienen una evolución continua del tiempo, e infinitos valores posibles de amplitud (es una magnitud física). Son señales continuas y no cuantizadas. Ejemplo: la voz humana.
- Las **señales digitales** tienen el eje vertical discretizado: hay una cantidad finita de valores que puede tomar la función, y los momentos de cambio típicamente también son discretizados, pues las señales suelen seguir un cierto patrón temporal (hay un reloj, que dice cuándo una señal puede cambiar de un valor a otro).

Normalmente, la información que uno quiere enviar está codificada de manera digital, pero en el nivel físico la señal debe ser transformada en alguna modulación de la señal (una señal analógica), que estará sometida a ruido.

3.2.1. Ondas

Las señales analógicas viajan en forma de ondas electromagnéticas, por lo que su velocidad es un factor de la velocidad de la luz (el valor de la velocidad de transmisión depende del medio). Es decir, viajan muy rápido, pero cuando las distancias son realmente grandes (por ejemplo, cuando uno se quiere conectar a un servidor del otro lado del mundo), el **delay** empieza a jugar un papel importante.

Estas señales son oscilantes: tienen un **período** (cuánto tiempo dura un ciclo). La cantidad de veces por unidad de tiempo en que la onda completa un ciclo es lo que determina su **frecuencia**, lo cual a su vez limitará la velocidad a la que se puede transmitir la información.

Un ciclo de una onda tomará una determinada distancia en producirse; esto se llama **longitud de onda**.

Problemas: Las ondas pueden chocar con imperfecciones del medio físico, lo cual produce pérdidas de energía, atenuando la señal. También habrá fuentes de ruido que perturben la señal.

Hay ciertas propiedades de una onda que se pueden manipular (**modular**), como la amplitud, la frecuencia y la fase. Esto permite codificar valores binarios (bits), pues se puede especificar por ejemplo que una onda con cierta amplitud codifique un 0, y una con otra amplitud codifique un 1.

Dada una señal periódica, existen herramientas para transformarla en una sumatoria de señales más simples:

- Cualquier función periódica se puede descomponer en la llamada *serie trigonométrica de Fourier*, usando la frecuencia fundamental de la onda.

Onda Cuadrada: se puede representar como una serie infinita de senoides armónicamente relacionadas.

- Sin embargo, al transmitir una onda (por ejemplo una cuadrada) por ciertos medios físicos, el fenómeno de filtrado produce que se pierdan ciertas partes de la misma.
- ¿Cómo saber entonces si la información transmitida en esa onda llega de manera íntegra al destinatario?
- La matemática propuesta por Shannon vincula el ruido, el ancho de banda (distancia entre frecuencia mínima y máxima que se puede transmitir), las señales y la información contenida en un mensaje.
- Una onda cuadrada se puede representar, usando la serie de Fourier, como infinitas ondas (armónicos), por lo cual sería necesario un canal ideal para transmitir sin pérdida, lo cual no es factible para distancias considerables.

3.3. Ancho de Banda

Es la distancia entre la frecuencia mínima y la frecuencia máxima que se puede transmitir por un canal determinado. Es decir que es el ancho frecuencial en el que se puede usar el canal sin que se deteriore (demasiado) la señal.

Filtrado: es un fenómeno por el cual el medio físico deja pasar ciertas frecuencias de una onda, pero otras no.

Para determinar ese mínimo y máximo, se toma una **frecuencia de corte**, según cuál sea la frecuencia con la que se produce una atenuación de 3 decibeles.

Las frecuencias dentro del área comprendida por el ancho de banda son aquellas en las que se produce menor atenuación debido al medio de transmisión (filtrado).

3.4. Medios de Transmisión

En este nivel, es claro que la comunicación se da entre nodos (*next hop*) y no *end-to-end*, ya que el cable que conecta dos nodos es lo que lleva a cabo la comunicación.

Esto no significa que el siguiente nodo en la comunicación siempre sea conocido por el usuario (puede ser algún router perteneciente al proveedor del servicio de internet, por ejemplo).

Los medios de transmisión física son las tecnologías de acceso (e.g. fibra óptica, cable coaxial, Wi-Fi, etc.). Hay dos grandes categorías de medios:

1. Medios guiados (cableados): par trenzado de cobre, cable coaxial, red eléctrica, fibra óptica. Las ondas están confinadas dentro del cable, normalmente con alguna protección para reducir el ruido, amplificadores de señal cada cierta distancia.
2. Medios no guiados (inalámbricos): radio, microondas, ondas infrarrojas, láser, Wi-Fi, Li-Fi (usar pulsos de luces LED para codificar información).

3.5. Red Telefónica

Es una red de onda guiada que le da forma a muchas de las redes que se utilizan más hoy en día.

3.5.1. Estructura del Sistema Telefónico

- **Objetivo:** transmitir la voz humana de manera reconocible.
- Se organiza en una **jerarquía multinivel** con redundancia (más de un camino para ir de un nodo a otro).
- Está compuesto por *local loops* (pares trenzados, señalización analógica), troncales (fibra óptica o microondas, señalización digital) y oficinas de conmutación. Esto se puede ver en la figura 5.

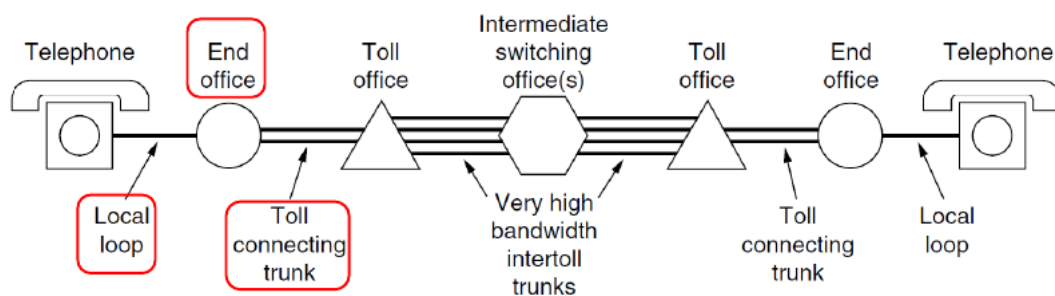


Figura 5: Ruta típica seguida para una llamada de larga distancia

3.5.2. Multiplexación

Por diferentes motivos (e.g. económicos), las compañías telefónicas desarrollaron formas de hacer que diferentes comunicaciones telefónicas vayan por un mismo troncal físico. Esto se conoce como **multiplexación**.

Hay dos tecnologías básicas usadas para esto:

- **TDM** (Time Division Multiplexing): los usuarios toman turnos, durante los cuales obtienen el **ancho de banda completo** por un **período de tiempo acotado**. Es como una fila de un peaje.
- **FDM** (Frequency Division Multiplexing): el espectro de frecuencias se divide en canales de **ancho de banda acotado**, usados a **tiempo completo y exclusivo** por cada usuario. Es como una avenida con varios carriles.

Ambas deben estar fuertemente administradas, y son completamente determinísticas en cuanto a la división de recursos.

FDM se utiliza aún, requiere circuitería analógica no trivial. En cambio, TDM puede manejarse con electrónica digital, por lo que en los últimos años se usó más.

Como el *local loop* produce señales analógicas (que transportan lo hablado por las partes de la conversación), hay que hacer una conversión analógico-digital en la *end office*, donde todos los *local loops* se combinan sobre los troncales. Esta conversión es la manera en la que se pueden digitalizar múltiples señales de voz (analógicas) para combinarlas en un único troncal digital.

Multiplexación por Longitud de Onda (WDM): es una técnica de FDM aplicada en sistemas ópticos (*dividir por colores*). Mezcla electrónica y óptica a niveles microscópicos, y se sigue utilizando hoy en día.

Taxonomía de las redes

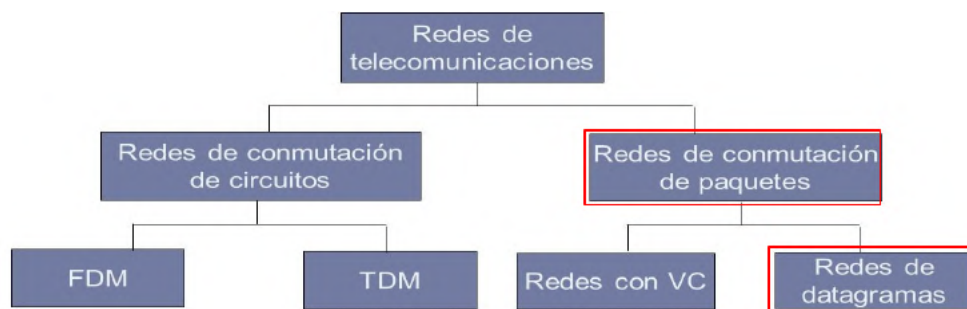


Figura 6: FDM y TDM pertenecen al *mundo* de la conmutación de circuitos.

Como se puede ver en la figura 6, las redes de conmutación de paquetes se dividen en dos grandes categorías:

- Redes de circuitos virtuales (VC): brindan un servicio orientado a conexión (e.g. X.25, ATM).
- Redes de datagramas (Internet): brindan un servicio sin conexión. Sin embargo, a nivel de transporte el servicio puede ser tanto orientado a conexión (TCP) como sin conexión (UDP).

Multiplexación estadística:

- Usada en las redes de **conmutación de paquetes**.
- “Estadística”: lo **no determinístico** es cuándo llega un nuevo paquete de información, cuánto mide ese paquete, y de qué usuario viene.
- División del tiempo **bajo demanda**: el canal está abierto para todos, el que llega primero lo usa.
- Los paquetes de distintas fuentes *comparten* el enlace a tiempos distintos.
- Los paquetes se *encolan*, y *compiten* por el enlace cuando éste no está disponible.
- Esto no permite dar tantas garantías como otros sistemas de multiplexación, al tener *reglas de juego* tan laxas.
- Cuando hay demasiados paquetes encolados y ocurre overflow, se dice que hay **congestión** en los buffers de entrada y de salida de los nodos de red (switches, routers, etc.).

3.6. Teorema de Muestreo

Recordar que usando la serie infinita de Fourier, es posible descomponer funciones periódicas (como las señales analógicas) en sumatorias de componentes, en senos y cosenos.

Si a estas sinusoides se las “muestrea” (*muestrear* es *sacarle fotos* a la onda cada cierto tiempo), el caso más crítico de muestreo será aquella componente de **mayor frecuencia** f_{max} , es decir la de menor período: como es la que más rápido oscila, será la que requiera un muestreo más frecuente.

El **Teorema de muestreo** de H. Nyquist dice que si se quiere reconstruir una señal de componente frecuencial máxima f_{max} , habrá que muestrearla según $f_s > 2 \times f_{max}$. Esta f_s se denomina **frecuencia de muestreo** (*sampling*).

Es decir, hay que *sacarle fotos* a la onda con una frecuencia de *al menos el doble de la frecuencia* de la componente más rápida de la señal, para que no haya pérdida de información respecto a lo enviado por el transmisor.

3.7. Conversión Analógico-Digital

La conversión consta de dos etapas:

1. Se muestrea la señal al doble del ancho de banda de la misma (siguiendo lo visto en el teorema de muestreo), obteniendo un tren de **pulsos de amplitud variable** (PAM).
2. Se cuantifican las muestras, aproximándolas con un entero de **n bits**. Esto introduce un error de cuantificación.

Canal PCM (Pulse Code Modulation): Las señales analógicas son digitalizadas por un **CODEC** (COder-DECoder), que produce símbolos de 8 bits por muestra. Este CODEC toma 8000 muestras por segundo (4 kHz de ancho de banda, lo *suficiente* para transmitir la voz humana de manera entendible).

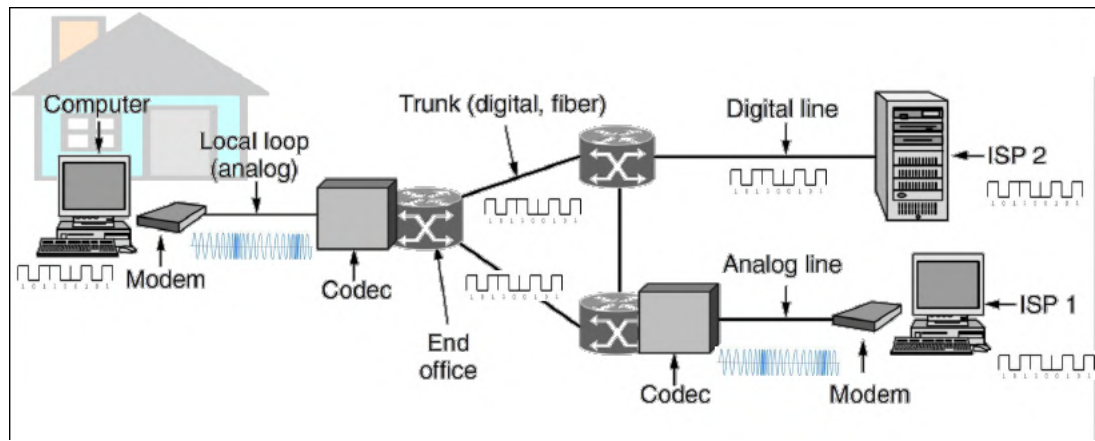


Figura 7: En una comunicación entre computadoras, se utilizan señales tanto analógicas como digitales, por lo que es necesaria la conversión, realizada por modems y codecs.

3.8. Modulación

Definición: la **modulación** es el proceso de variación de cierta característica de una señal sin mensaje (llamada **portadora**) de acuerdo con una señal mensaje (llamada **moduladora**).

La **velocidad de modulación** se define como el número de cambios de señal por unidad de tiempo, y se expresa en *baudios* (símbolos por segundo).

La **velocidad de transmisión** es la velocidad de modulación multiplicada por el número de bits representados por cada símbolo, y se expresa en bits por segundo.

Hay cuatro tipos de modulación, en base a qué tipo de señales se tenga:

1. Moduladora Analógica y Portadora Analógica
2. Moduladora Digital y Portadora Analógica
3. Moduladora Analógica y Portadora Digital
4. Moduladora Digital y Portadora Digital

3.8.1. Moduladora Analógica y Portadora Analógica

Cuando tanto la señal portadora como la modulante son analógicas, hay dos maneras de modular la portadora:

- Onda modulada en **frecuencia**: valores altos de la señal modulante implican una frecuencia baja, y viceversa (o al revés también).
- Onda modulada en **amplitud**: valores altos de la señal modulante implican una amplitud alta, y viceversa (o al revés también).

3.8.2. Moduladora Digital y Portadora Analógica

Este es el caso de la transmisión de datos digitales a través de la red de telefonía.

Hay varias maneras de hacerlo:

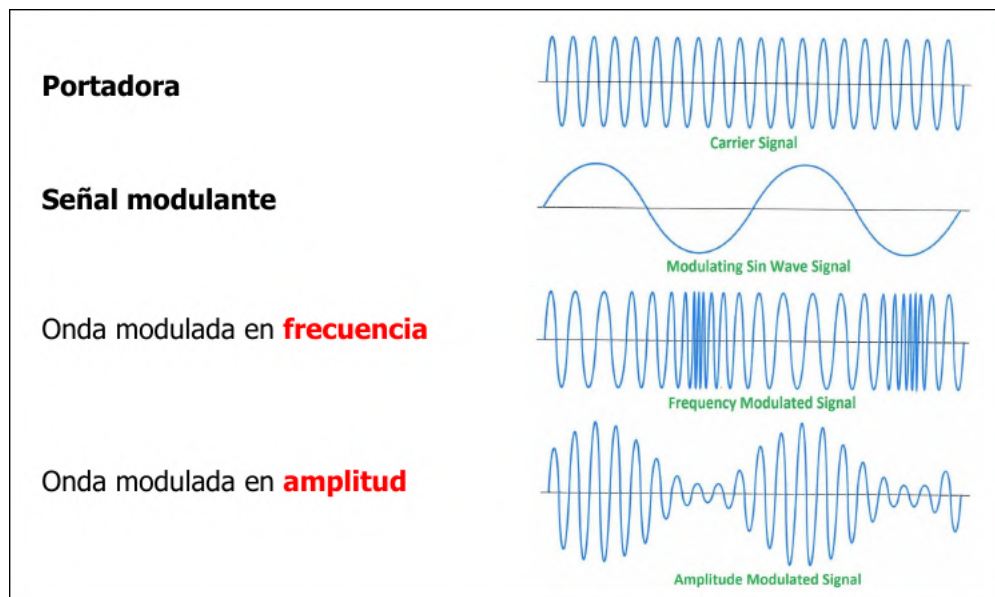


Figura 8: Modulación en frecuencia y en amplitud.

- Desplazamiento de Amplitud (**ASK**): los valores binarios se representan mediante dos amplitudes diferentes de la señal portadora.
- Desplazamiento de Frecuencia (**FSK**): los valores binarios se representan mediante dos frecuencias diferentes de la señal portadora.
- Desplazamiento de Fase (**PSK**): los valores binarios se representan mediante dos fases diferentes de la portadora.
- Mixtas.

Para conseguir un uso más eficaz del ancho de banda, se utiliza la **modulación multinivel**; esto es, cada elemento de la señal transmitida representa más de un bit.

3.8.3. Moduladora Analógica y Portadora Digital

Hay diferentes maneras de hacerlo, por ejemplo:

- Modulación por Impulsos Codificados (MIC, o **PCM**): observan la señal y sacan su valor.
- Modulación Delta (**DM**): codifica sólo las diferencias, y no en el valor de la señal como tal. Esto a veces es más eficiente.

3.8.4. Moduladora Digital y Portadora Digital

Los datos binarios se transmiten codificando cada bit de datos en cada elemento de la señal. También hay varios métodos:

- No retorno a cero (**NRZ**): consiste en utilizar una tensión negativa para representar un 0, y una positiva para representar un 1. El inconveniente principal con esto es que, para secuencias largas sin cambios, se pierde el sincronismo.

- No retorno a cero con inversión de unos (**NRZI**): los datos se codifican mediante la presencia o ausencia de una transición al principio del intervalo de un 1. Esto soluciona la mitad del problema mencionado para NRZ: se soluciona para muchos 1 consecutivos, pero no para muchos 0.
- Manchester (**Bifase**): se codifica mediante una transición en la mitad del intervalo de duración del bit; de bajo a alto es un 1, de alto a bajo es un 0. Se usa en el protocolo Ethernet.
- Manchester Diferencial (**Bifase diferencial**): la codificación de 0 se representa por la presencia de una transición al principio del intervalo del bit, y un 1 mediante la ausencia de transición.

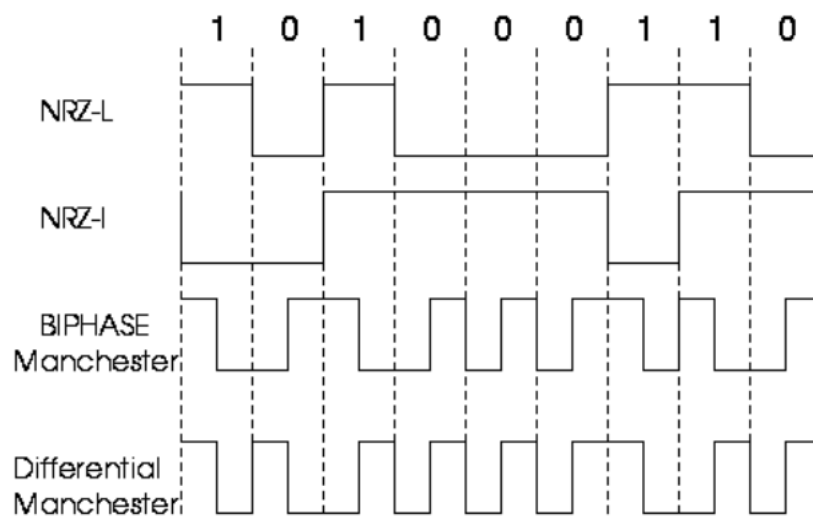


Figura 9: Los diferentes métodos de modulación con portadora y moduladora digitales, para una misma secuencia de datos.

3.8.5. Capacidad de Canal y Modulación

Todos los posibles métodos de modulación de señales están limitados por el teorema de capacidad de canal de Shannon, por más sofisticados que sean.

3.8.6. Bit Error Rate (BER) y Modulación

El precio a pagar en las modulaciones de orden superior, por la mejora en la velocidad de transmisión, es una mayor tasa de errores.

Es decir, cuanto más sofisticada es la tecnología de modulación (más bits por cada símbolo), mayor será el BER.

La clave está en hallar un equilibrio entre las mejoras en capacidad de canal (aún limitadas por el teorema de Shannon) y las mayores tasas de error.

3.9. Redes Inalámbricas

- En cualquier medio (guiado o no), la **intensidad de la señal disminuye con la distancia**. Esto toma mayor relevancia en medios inalámbricos.

- Las **fuentes de ruido son más impredecibles** que en medios guiados, lo cual lleva a tasas de errores más elevadas.
- El acceso al medio es **compartido** (multiacceso): se comparte un espectro de frecuencias se se propaga por el aire.
- En dispositivos móviles, el consumo de **energía** es un nuevo desafío.
- La potencia con la que se puede transmitir señal es **regulada** por distintas organizaciones según el territorio (e.g. en Argentina, ENACOM).
- Hay bandas del espectro electromagnético en las que se requiere **licencia** para transmitir (AM, FM, TV, etc.) y otras no licenciadas. En las bandas **no licenciadas**, se limita la potencia de transmisión, con lo cual se limita la distancia que puede recorrer la señal, y se aumenta la magnitud de las interferencias con otros dispositivos.
- Cuando el espectro es compartido por muchas aplicaciones y dispositivos, surge la idea de usar un **espectro expandido** (spread-spectrum).

3.10. Fuentes

- Rodrigo Castro. Teoría de las Comunicaciones, Clase Teórica 1. Primer cuatrimestre, 2020.
- Rodrigo Castro. Teoría de las Comunicaciones, Clase Teórica 2. Primer cuatrimestre, 2020.
- Esteban Lanzarotti. Teoría de las Comunicaciones, Clase Práctica 1. Primer cuatrimestre, 2020.
- Claudio Righetti. Teoría de las Comunicaciones, Clase Teórica 10. Segundo cuatrimestre, 2021.

4. Nivel de Enlace: Protocolos Punto a Punto

4.1. Fundamentos

La idea ahora es asumir que todo lo que ocurre a nivel inferior está solucionado y funciona correctamente. Es decir, los aspectos de la capa física no interesan en este punto.

Se dispone de un *caño* serial, sin desordenamiento, pero sí sometido a ruido y fallas.

4.2. Encapsulamiento (framing)

Hay varias opciones para separar los frames en un tren de bits:

- Largo fijo
- Largo especificado en el encabezado
- Delimitadores de frame (*bit-stuffing*).

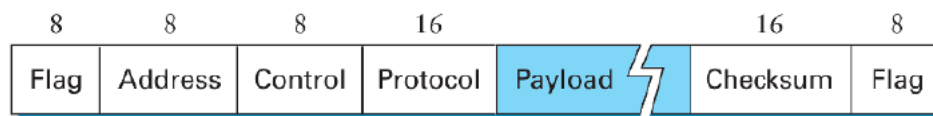


Figura 10: Ejemplo de un frame para un protocolo punto a punto (PPP). Por un lado la información (*payload*) y por otro lado otros campos con información de control.

4.2.1. Eficiencia de un Frame

Es la proporción de datos en un frame respecto del total del mismo (el resto es la información de control).

$$\eta_{frame} = \frac{|datos|}{|frame|}$$

4.3. Tipos de Servicio

1. Sin conexión y sin reconocimiento: los datos se envían sin necesidad de saber si llegan con errores o no.
2. Sin conexión y con reconocimiento: los datos se envían y se asegura la correcta recepción sin errores mediante el **aviso explícito** (ACKs).
3. Orientado a conexión: además de asegurar la ausencia de errores en la recepción de los datos, se mantiene un **estado de conexión** (una *sesión*).

4.4. Detección y Corrección de Errores

Redundancia: m bits (**datos**) + r bits (**redundancia**) = n bits (**codeword**)

Se necesita:

- $e + 1 \leq d$ para poder **detectar**.
- $2e + 1 \leq d$ para poder **corregir**.

donde d es la mínima *Distancia de Hamming* entre todas las codewords de un código, y e es la cantidad de bits erróneos en una transmisión.

Para garantizar la **confiabilidad**, es necesario poder hacer **retransmisiones**. Estas pueden ser:

- Implícitas: cuando ocurre un time-out, se asume que el dato se perdió.
- Explícitas: existen mensajes de control específicos para pedir la repetición del envío.

4.5. Transmisión Confiable

4.5.1. Stop & Wait

- El emisor envía un frame, y espera a recibir la confirmación (ACK) del receptor.
- *Cada frame* debe ser **reconocido por el receptor**.
- Se usa el **time-out** para dejar de esperar la llegada de un ACK una vez pasado cierto tiempo.
- Aparece el **problema de las reencarnaciones**: no saber a qué envío de frame corresponde un ACK. Para solucionar esto, es necesario **secuenciar** los frames: numerarlos unívocamente. En el caso de Stop & Wait, es necesario secuenciar al menos dos frames (basta un bit para eso).
- Hay un **tiempo de bloqueo** mientras se esperan las confirmaciones. Esto es una subutilización del canal, lo cual implica una baja eficiencia.

En la figura 11, se muestran cuatro escenarios diferentes que pueden ocurrir al utilizar Stop & Wait:

- a) Es el *caso feliz*; el emisor envía un frame, y el receptor le manda el ACK confirmando recepción. El ACK es recibido por el emisor antes de que ocurra el time-out.
- b) El emisor envía un frame, pero éste no le llega al receptor. Por lo que cuando ocurre el time-out, se reenvía el frame, que esta vez sí llega y es confirmado por el ACK.
- c) El emisor envía un frame, y el ACK no le llega, por lo que vuelve a enviar el mismo frame cuando ocurre el time-out; esta vez sí obtiene el ACK.
- d) El emisor envía un frame, el receptor lo recibe y envía el ACK, pero ocurre el time-out antes de que el emisor se entere. Por lo tanto, vuelve a enviar el mismo frame, pero el ACK del envío anterior llega justo después. Por lo tanto, el receptor asume que el siguiente frame que reciba será distinto del que acaba de confirmar. Este es el **problema de las reencarnaciones** (esto también pasa en el c)).

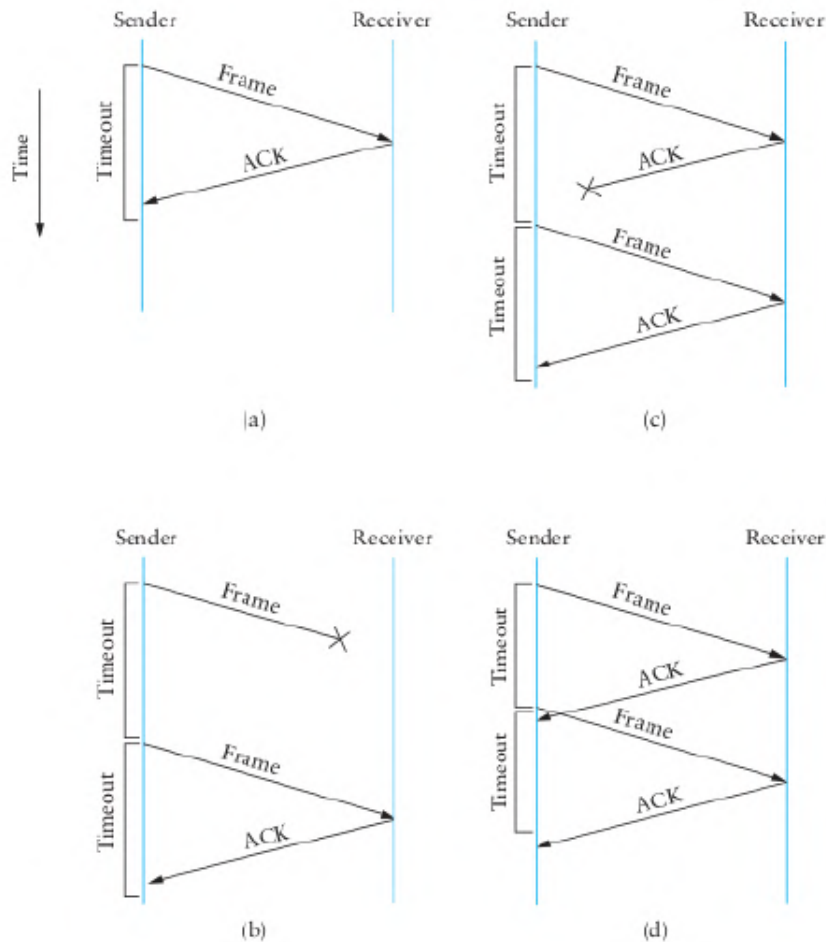


Figura 11: Protocolo Stop & Wait

4.5.2. Eficiencia de un Protocolo

La eficiencia de un protocolo es una expresión que sirve para representar cuánto tiempo se está transmitiendo con respecto al tiempo que se está esperando por las confirmaciones.

$$\eta_{proto} = \frac{T_{tx}(F)}{RTT(F)}$$

donde:

- $T_{tx}(F)$ es el tiempo de transmisión de un frame.
- $RTT(F)$ es el *roundtrip time* (tiempo de ida y vuelta) de un frame. Es decir, el tiempo que tarda en llegar el frame al receptor, y llegar la confirmación al emisor. En general, $RTT = Delay \times 2$.

Aumentar la eficiencia de un protocolo es estar **lo menos posible bloqueado esperando**. Una estrategia posible para esto es **enviar varios frames seguidos**, sin esperar ACKs para cada uno.

Para eso, aparece el concepto de **ventana de frames**: en una ventana se envía una cierta cantidad de frames. Esto resulta en una definición diferente para la eficiencia:

$$\eta_{proto} = \frac{T_{tx}(V)}{RTT(F)}$$

donde esta vez $T_{tx}(V)$ es el tiempo de transmisión de una ventana.

Cuántos frames enviar en la ventana corresponde al diseño del protocolo.

4.5.3. Sliding Window

Motivación: mantener lleno el canal, en lo que tarda en venir el ACK, para mejorar la eficiencia.

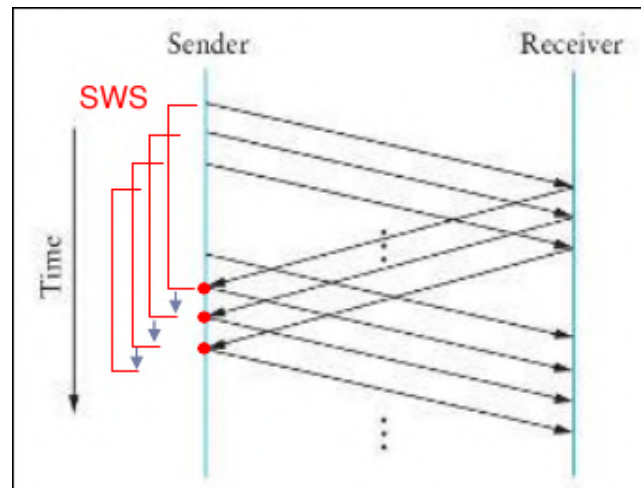


Figura 12: Protocolo Sliding Window

Se envían los frames en una **ventana de emisión**, cuyo tamaño es:

$$SWS = \frac{V_{tx} \times RTT}{|Frame|} frames$$

Se envían nuevos frames siempre que se verifique que: $\text{ÚltimoFrameEnviado} \leq \text{ÚltimoFrameReconocido} + SWS$.

El receptor puede bufferear los frames que recibe o no, dependiendo del esquema de ACKs:

- ACKs acumulativos (figura 13).
- ACKs selectivos (SACK) (figura 14).

Así, el tamaño de la **ventana de recepción** es:

$$RWS = \begin{cases} SWS & \text{si hay SACK,} \\ 1 & \text{si no} \end{cases}$$

Ante un frame que no le llegue correctamente al receptor, hay varias estrategias que se pueden seguir:

- **GoBackN**: el receptor no necesita un buffer para frames. No envía el ACK del frame que no llegó, y descarta todos a partir de ese; por lo tanto, el emisor debe reenviar todos los frames a partir del que no llegó. Notar que si el frame que no llega es el primero, es la misma situación que Stop & Wait.

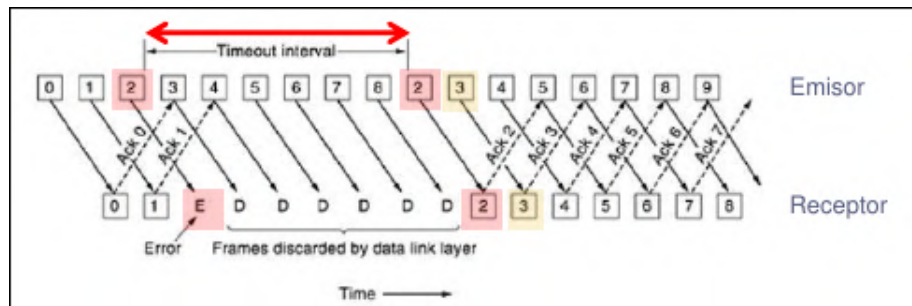


Figura 13: ACKs acumulativos

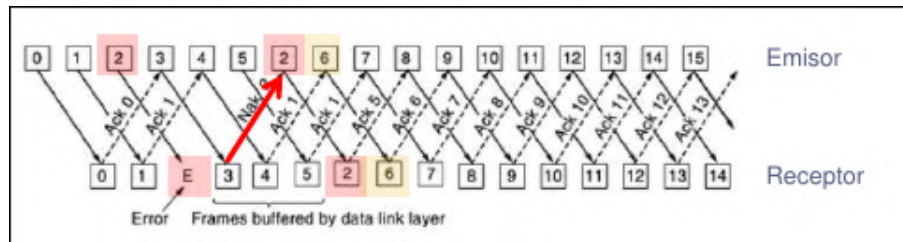


Figura 14: ACKs selectivos (SACK), de tipo Negative ACK

- **Selective ACK (SACK)**: ahora no sólo el emisor tiene su ventana de frames, sino que también el receptor la tiene (y del mismo tamaño que la del emisor). El receptor envía un ACK selectivo con el frame que no llegó, pero se guarda los recibidos correctamente, por lo que el emisor envía sólo el que falló. Esto sólo se puede usar si hay memoria para bufferear los frames. Hay una especie de variante del problema de las reencarnaciones que puede darse al usar SACK, en donde se corre la ventana y aparecen varias versiones de un frame pero de distintas ventanas. Para distinguir **reencarnaciones** es necesario que la cantidad de frames unívocamente identificables sea mayor o igual que $SWS + RWS$ (es decir la cantidad de frames a secuenciar).

4.6. Delay, Propagación y Transmisión

Delay: es el tiempo total que se tarda en enviar información de un punto a otro:

$$Delay = T_{tx} + T_{prop} + T_{queue} + T_{proc}$$

donde:

- $T_{tx} = |datos|/V_{tx}$ es el **tiempo de transmisión**. Es decir, el tiempo requerido para *empujar* todos los bits de un paquete a través del medio de transmisión. Es significativo para enlaces de baja velocidad, o frames de gran tamaño.
- $T_{prop} = distancia/V_{prop}$ es el **tiempo de propagación**. Es decir, el tiempo transcurrido en, una vez que cada bit es *empujado* en el medio, su propagación hasta el final del trayecto físico. Es significativo para enlaces muy distantes.
- T_{queue} es el **tiempo de encolamiento**. Es decir, el tiempo en que el paquete espera en un buffer hasta ser transmitido. Depende de la congestión, por lo que puede ser nulo o enorme.
- T_{proc} : es el **tiempo de procesamiento**. Es decir, el tiempo requerido para analizar el encabezado y decidir a dónde enviar el paquete, pudiendo incluir también la verificación

de errores. Suele ser de unos pocos microsegundos o menos, por lo que en la práctica se tomó como nulo.

- V_{tx} es la **velocidad de transmisión**.
- V_{prop} es la **velocidad de propagación**, que depende de la distancia del medio físico, y es cercana a la velocidad de la luz en la mayoría de los casos.

4.7. Capacidad de Volumen

Es la cantidad de bits que entran en el medio desde que se envía el primer bit hasta que éste llega al receptor, es decir la cantidad de bits que *entran* en un canal a la vez:

$$C_{vol} = Delay \times V_{tx}$$

Sin embargo, en el caso de un protocolo punto a punto en la capa de enlace, se debería calcular cuántos bits entran en el canal **hasta recibir el primer ACK**, es decir usando $2 \times Delay = RTT$. Esto es para aprovechar mejor el canal.

Es decir, se calcularía la capacidad de volumen como:

$$C_{vol} = RTT \times V_{tx}$$

Usando esto, junto con la cantidad de bits de cada frame, se puede calcular cuántos frames se deben enviar para aprovechar al máximo el canal, es decir el tamaño óptimo para una ventana.

4.8. Fuentes

- Rodrigo Castro. Teoría de las Comunicaciones, Clase Teórica 2. Primer cuatrimestre, 2020.
- Paula Verghelet. Teoría de las Comunicaciones, Clase Práctica 2. Primer cuatrimestre, 2020.

5. Nivel de Enlace: Protocolos de Acceso Múltiple

5.1. Acceso a Medios Compartidos

La idea es **controlar el acceso** a los medios compartidos de manera tal que haya la **menor cantidad de intervención humana** posible en el proceso. Es decir, se busca no tener la figura de un administrador que tenga que solucionar manualmente las cosas.

Tanto TDM, FDM, WDM y CDMA (Code Division Multiple Access) comparten una característica: debe estar decidido *a priori* qué usuario está usando qué parte del tiempo, frecuencia, etc. en cada momento. Para eso es necesario saber de antemano cuántos usuarios tendrá el sistema. Esto requiere un administrador dedicado a una red con características **estáticas, rígidas**. Esto no escala de manera automática, por lo que no cumple la idea mencionada arriba.

Esto no significa que estas técnicas no se usen: se utilizan frecuentemente en las **redes troncales** (backbones), que no tienen una cantidad constantemente cambiante de nodos como sí puede tener algo como una red LAN, o Wi-Fi.

Una alternativa a esto, para redes donde la cantidad de nodos es desconocida y cambiante, es la **contención estadística**. Esto se refiere a sistemas en los cuales varios usuarios comparten un canal común, de modo tal que *puede dar lugar a conflictos* conocidos como **sistemas de contención**. Estos conflictos son aceptados y/o manejados.

5.1.1. Problema de Acceso

Si hay varios nodos que usan un medio físico compartido, la **simultaneidad** de transmisión **no es posible** (no pueden transmitir todos a la vez).

Para esto aparecen los **MAC Protocols** (Medium Access Control), protocolos que buscan **maximizar, en promedio, el número de éxitos** en los intentos de comunicación, y asegurar la *igualdad de oportunidades* (en promedio) entre todos los nodos *competidores*.

En estos casos, el control es **descentralizado**, y surge la necesidad de un esquema de direccionamiento y de controlar el acceso (se podría usar FDM, TDM, etc. pero no es la idea, justamente porque se busca un control descentralizado y no administrado).

Ejemplos: Aloha, Ethernet, Wi-Fi, Token Ring.

5.2. Ethernet (IEEE 802.3)

Es una tecnología que ha evolucionado mucho con el tiempo. Por ejemplo, en la tasa de transmisión permitida por los cables, o el alcance máximo.

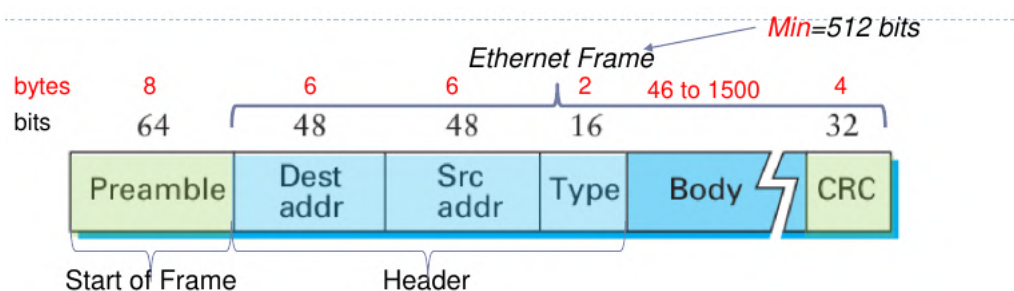


Figura 15: Frame de Ethernet.

Un host recibe **frames** que estén destinados a:

- su dirección (unicast).
- la dirección broadcast (FF:FF:FF:FF:FF:FF).
- una dirección multicast (de estar suscrito).
- cualquier frame (si se activó el modo promiscuo).

5.3. Mecanismo de Acceso: CSMA/CD

CSMA/CD (Carrier Sense Multiple Access with Collision Detection), es decir, acceso múltiple con sensado de portadora y detección de colisiones, es un algoritmo de control de acceso a un medio compartido.

Utiliza el sensado de portadora para determinar si hay nodos transmitiendo: cuando un host tiene datos para enviar, **sensa el medio** compartido.

- Si el medio está **libre**, el host transmite.
- Si el medio está **ocupado**, no puede enviar porque habría una colisión. Hay distintas formas de reaccionar:
 - 1-persistente: espera a que se libere, y transmite (es el caso de Ethernet 802.3).
 - p-persistente: espera a que se libere, y transmite con probabilidad p .

El uso de un **componente azaroso** (en la p-persistencia) tiene sentido porque si hay varios hosts esperando a que se libere el medio, y todos intentan transmitir ni bien éste se libera, va a ocurrir una colisión. Imponer una probabilidad para transmitir reduce las probabilidades de colisiones.

Este algoritmo es de categoría **half-duplex**: la lógica de recepción está establecida en el sensado para detectar colisiones. Es decir, **no se puede enviar y recibir a la vez** (eso sería full-duplex).

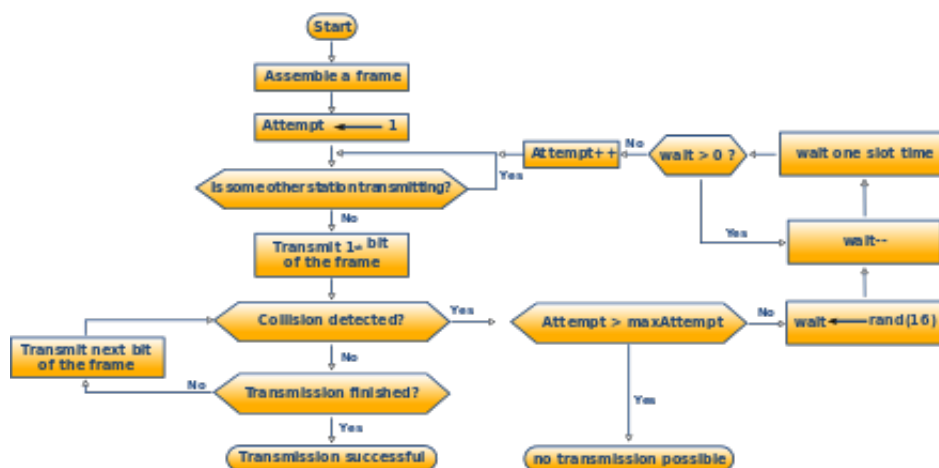


Figura 16: Algoritmo simplificado de CSMA/CD, con lógica de retransmisión para resolver una colisión.

5.3.1. Colisiones

Si dos hosts envían un frame a la vez, se producirá una colisión. Por lo tanto, se necesita control sobre los envíos, para **saber si llegaron sin colisionar**.

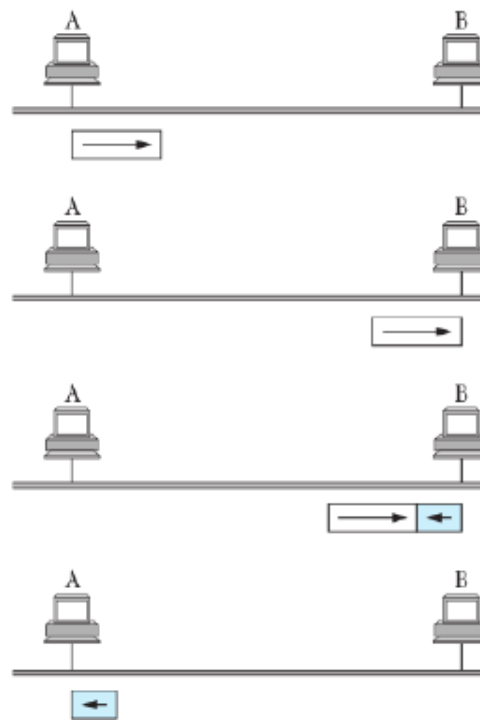


Figura 17: Escenario de peor caso.

En la figura 17 se puede ver un posible escenario de peor caso: si se tiene a los hosts A y B en un medio compartido, y cuando A sensa el medio, este está libre, entonces A empieza a transmitir. Como la velocidad de transmisión es finita, puede ocurrir que B sense el medio y también lo detecte como libre, pues todavía no *empezó a llegar* la transmisión de A. Entonces, B también transmitirá, y ocurrirá una colisión.

El objetivo no es evitar que ocurran este tipo de colisiones, sino poder detectar que ocurrieron. Para esto, se impone un **largo mínimo de frame: se envía hasta saber que no hubo colisión**.

En el caso de la figura 17, el host A estará transmitiendo una cantidad suficiente de tiempo tal que el largo del mensaje sea suficiente para alcanzar al host B.

El host A está escuchando su propio mensaje, por lo que si el host B inyecta una señal en el medio mientras A transmite, A podrá detectar la suma de las dos señales. Si detecta un nivel de señal del doble de lo estándar, será indicador de que se produjo una colisión.

Como la distancia entre hosts no es nula, y la velocidad de propagación de las ondas por el medio es finita, será necesario transmitir una cierta cantidad mínima de información, por lo que es necesario imponer un tamaño mínimo para las tramas (frames) del protocolo.

Cuando ocurre una colisión, antes de retransmitir, el host envía una **jam sequence**, una secuencia de bits desordenados, que sigue cierto patrón para indicar a todos los participantes del medio que hubo una colisión, para que ninguno transmita en ese momento.

5.3.2. Retransmisiones

Cuando se detecta una colisión, los dos hosts involucrados tienen que retransmitir. Esto se puede hacer inmediatamente (esto va a generar una nueva colisión), o luego de un tiempo, que puede ser fijo o aleatorio.

Lo que se implementa para este fin es el mecanismo de **exponential backoff**: se subdivide el tiempo en *slots* de alguna duración fija, que serán los que determinen la demora (backoff) que ocurrirá antes de la siguiente retransmisión.

Entonces, el host que tenga que retransmitir elige un *slot* entre 0 y $2^k - 1$, siendo k la cantidad de intentos de retransmisión hasta el momento. Así, el host esperará *slot* veces el RTT antes de sensar para retransmitir.

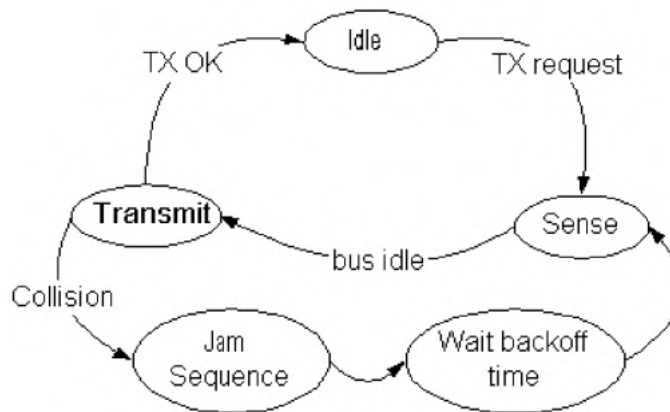


Figura 18: Esquema de estados de un transmisor en CSMA/CD.

5.3.3. Performance

Una manera de estudiar la eficiencia de estos protocolos es con gráficos que muestren el **goodput** G (proporción de transmisiones exitosas por unidad de tiempo) en función de la **carga ofrecida** S (número de intentos de transmisión por unidad de tiempo, mide la intensidad con la que se está intentando usar el medio).

$$S = G \times P(\text{colision})$$

Si hay mucha carga ofrecida (los hosts son muy *insistentes* con sus intentos de transmisión) en un contexto con muchas colisiones, el goodput será bajo (aumenta la probabilidad de colisionar). Esto se puede ver en las curvas del gráfico en la figura 19, pues todas tienden a decrecer.

En la curva inferior del gráfico se puede ver el protocolo Aloha (N. Abramson, 1970), precursor de la idea de CSMA. En este protocolo, un host retransmite si no recibe un ACK para su transmisión. Su eficiencia máxima de 18 % (de cada 100 bits que se envían, 18 llegan correctamente) se alcanza con un valor de $G = 0.5$, es decir que se reintenta una vez cada dos ventanas.

La curva del CSMA/CD 1-persistente (incorporado por Ethernet 802.3) mejora la performance a partir de la introducción de la detección de colisiones. Este protocolo alcanza una eficiencia máxima de alrededor del 50 % reintentando una vez por ventana de reintentos.

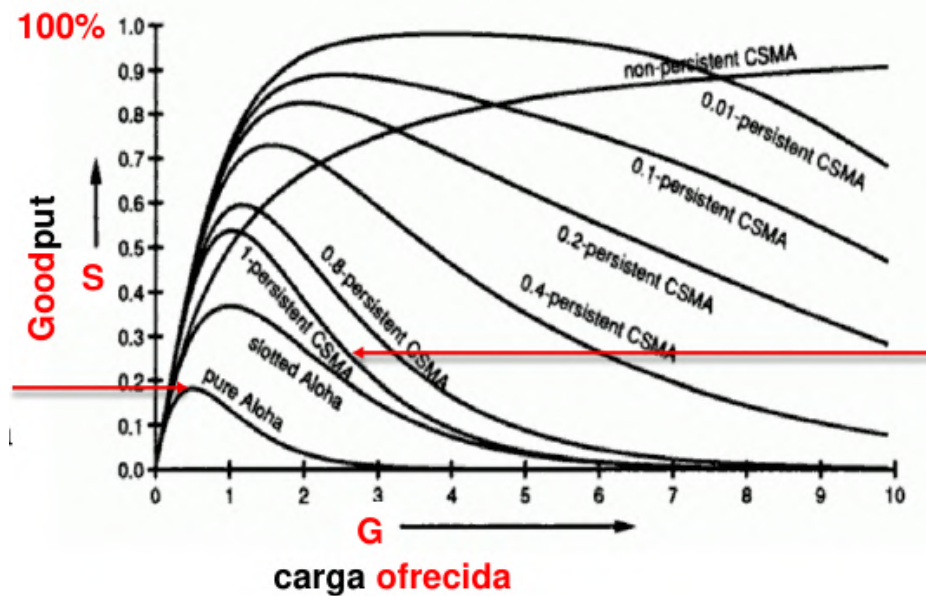


Figura 19: Goodput de diferentes algoritmos de acceso a medios compartidos, en función de su carga ofrecida.

Las versiones de CSMA p-persistente con valores de p más bajos obtienen picos de eficiencia más altos. Es decir, se mejora el goodput al establecer una probabilidad de transmisión más baja; pero esto *se paga* en que se espera más tiempo antes de transmitir: la tasa de envío será más baja (muchos menos errores en la transmisión, pero tardando mucho más tiempo). En resumen, hay un **trade-off** entre el delay y la cantidad de información neta enviada exitosamente.

En cuanto a la familia de protocolos CSMA, se puede concluir que es muy **flexible de implementar**, pero **escala muy mal** con la carga ofrecida G .

5.3.4. Ventajas y Desventajas

Ventajas:

- La detección de colisiones en redes LAN cableadas es fácil.
- El tiempo medio necesario para detectar una colisión es relativamente bajo.
- Puede ser empleado en sistemas de control de procesos continuos si la carga de tráfico de la red es baja (inferior al 20%).
- Ofrece un rendimiento mayor, en especial cuando existen pocas colisiones.

Desventajas:

- No es posible garantizar un tiempo máximo finito para el acceso de las tramas al canal de comunicación (problemas con real-time).
- No se puede usar con redes *half-duplex*, como Wi-Fi 802.11 (mientras una estación envía información es incapaz de escuchar el tráfico existente).
- Problemática en redes inalámbricas (problema de la terminal oculta).

5.4. Logical Link Control

La capa de enlace se puede dividir en:

- Medium Access Control (MAC): relacionado con el medio físico concreto.
- Logical Link Control (LLC): relacionado con lógica de control, independiente del medio físico (encapsula distintos tipos de medios, como Wi-Fi, Ethernet, etc.).

Así, los paquetes suelen tener contenido de MAC y contenido de LLC.

Recordar que la capa de enlace puede ofrecer tres tipos de servicios:

- Sin conexión y sin ACK (e.g. CSMA).
- Sin conexión y con ACK (e.g. protocolos punto a punto).
- Orientado a conexión (suelen usarse en una capa superior).

5.5. Local Area Network

5.5.1. Topologías de Red y Dominios

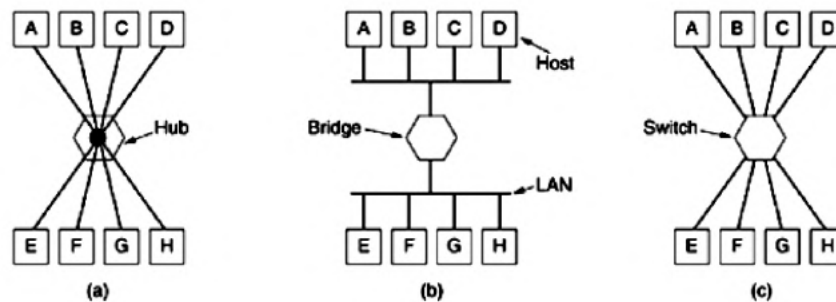


Figura 20: Diferentes topologías para una red local.

En la figura 20 se pueden ver tres topologías diferentes para redes de área locales (LAN, Local Area Network):

- En la a), los hosts están todos conectados a un **hub**: se comparte el medio físico directamente, por lo que hay un único dominio de colisión.
- En la b) se puede ver una evolución de ese modelo, en la que hay dos dominios de colisión diferentes (hay dos CSMA diferentes siendo utilizados), cada uno para un conjunto de nodos. Si se quieren comunicar nodos de diferentes dominios, lo hacen a través de un **bridge**. En vez de haber ocho hosts que colisionan, hay dos grupos de cuatro, para reducir la probabilidad de colisiones.
- En la c) se ve una red *switchheada*, donde el dominio de colisión desaparece: cada host se conecta con su cable (full-duplex) a un **switch**, y toda la información va a los buffers que tiene el switch, para que este haga el *dispatch* que corresponda. En este tipo de topología desaparecen las colisiones y CSMA/CD, convirtiéndose en un problema algorítmico relacionado con los buffers y el *dispatch*.

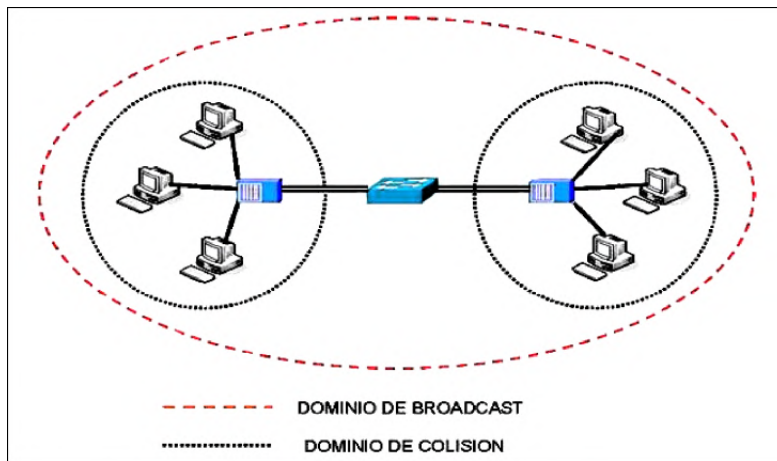


Figura 21: Dominio de colisión versus dominio de broadcast: el primero comprende a los nodos de las redes locales (conectadas a los hubs), pero el segundo abarca toda la red, dado que todos los nodos son alcanzables (algunas comunicaciones requerirán pasar por el bridge del centro).

5.5.2. LAN Extendida con 802.2

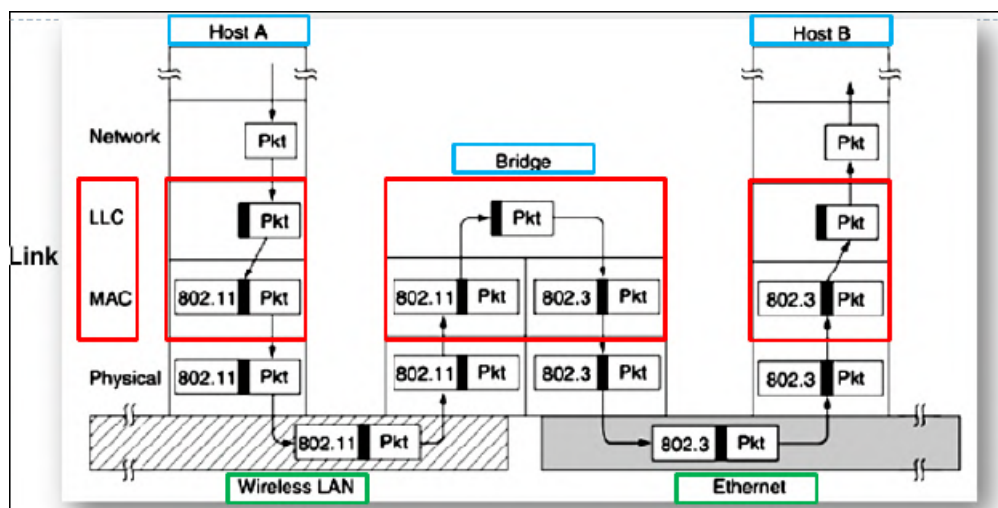


Figura 22: Ejemplo de una red LAN *extendida*, compuesta por dos hosts y un bridge.

En la figura 22, se puede ver dos hosts A y B con distintas tecnologías de capa física (Wi-Fi y Ethernet respectivamente), comunicados mediante un bridge. Por eso, los hosts no comparten dominio de colisión.

Como hay tecnologías distintas en los hosts, el nodo intermedio tendrá que *traducir* entre ellas. El bridge consume un paquete de una de sus dos *patas* (una en Wi-Fi y otra en Ethernet), este paquete sube por las capas, rescata el contenido del mensaje y le saca el encabezado correspondiente a la tecnología del host emisor, para ponerle el correspondiente a la del receptor.

Un bridge no es el único tipo de multiplexor que se puede usar. Los multiplexores se pueden categorizar según la capa o nivel en la que operan:

- Físico: repetidores y hubs.
- Enlace: bridges y switches.
- Red: routers.

La capa de enlace aparece dividida en MAC y LLC, donde LLC no tiene en cuenta la tecnología física presente, pero MAC sí debe incluir información en los paquetes para la detección de colisiones.

La conclusión es que:

- Las LANs pueden ser de varios tipos de tecnologías.
- Las estaciones deben compartir **esquema de direccionamiento**: hay un identificador único, a nivel de enlace, para cada nodo (MAC address).
- Esto permite que se puedan construir redes heterogéneas en cuanto a tecnologías, pero sigan siendo parte de la misma LAN sin un administrador centralizado.
- Usar estas LANs extendidas se utilizan por razones heterogeneidad, distancia, aislamiento, redundancia, seguridad...

5.6. Learning Bridges

El objetivo es escalamiento y flexibilidad: poder armar redes más grandes, conectadas por bridges/switches, y evitar que todos los nodos compitan por el medio con todos los demás, y sin necesitar un administrador para manejar las redes.

Para esto surge la idea de los **learning bridges**: los bridges *aprenden* en qué segmento de la red está cada host.

El comportamiento más básico para un bridge consistiría en: ante cualquier frame que le llegue al switch desde uno de los segmentos de la red, lo envía a todos los demás segmentos (inundación o **flooding**). Así se garantiza que el frame va a llegar a su receptor (en realidad llega a todos lados).

Esto no tiene mucho sentido dado que justamente se está usando un bridge para conectar las redes con el objetivo de reducir la posibilidad de colisiones.

Por lo tanto, lo que hacen los bridges es *aprender* en cuál de sus puertos (es decir, en cuál de los segmentos) se encuentra cada host. Entonces, cuando reciba un nuevo frame sólo lo enviará al segmento que corresponda (lo dejará salir por el puerto adecuado).

Para que este proceso de *aprendizaje* no requiera de un administrador, los bridges aprenden por su cuenta: **relacionan direcciones MAC con interfaces** (los puertos del bridge) en función del tráfico en la LAN. Esto significa que si un host nunca emite un paquete, el bridge nunca se enterará de dónde está.

Si el bridge no sabe en qué interfaz enviar el paquete, hará un flooding para asegurar que llegue correctamente.

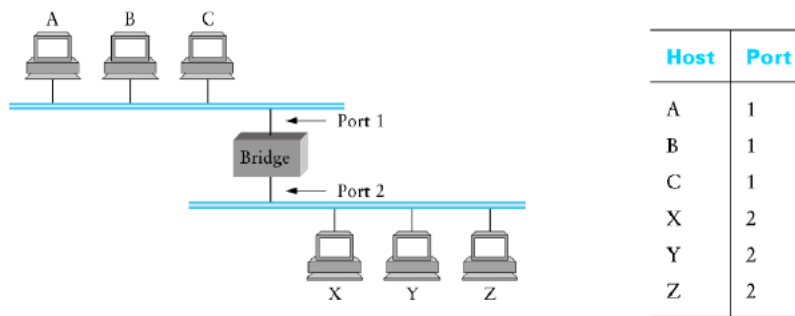


Figura 23: Ejemplo de un learning bridge y su tabla de reenvíos.

En resumen, los bridges:

- No reenvían cuando sea innecesario.
- Mantienen una tabla de reenvíos.
- Aprenden entradas de la tabla en base a la dirección de origen, y si faltan entradas, hacen inundación.
- Siempre envían los frames que sean *broadcasts*.

5.6.1. El Problema de las Topologías con Ciclos

El mecanismo de learning bridges funciona, pero también implica la aparición de nuevos problemas: **sólo funciona correctamente si no hay ciclos** en la topología de la red.

Una red LAN que crece dinámicamente sin administración va a terminar desarrollando ciclos, y no tiene sentido que haya alguien verificando si cada nuevo nodo generaría un ciclo. De hecho, a veces los ciclos son deseables, por motivos de redundancia.

En la figura 24 se puede ver un ejemplo de situación en la que se generan problemas en los learning bridges por culpa de los ciclos.

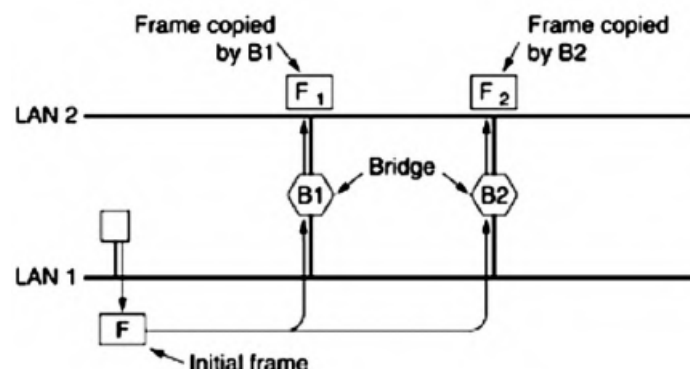


Figura 24: LAN extendida con ciclos.

Lo que ocurre aquí es que un nodo de la LAN 1 emite un frame en su segmento de red, y los dos bridges, B1 y B2, lo reciben, y aprenden que el nodo está en la interfaz inferior, pero reenvían el frame a la LAN 2, ya que no tenían información sobre a dónde enviar el frame.

Luego, hay dos copias diferentes del mismo frame en la LAN 2: la copia 1 será recibida por B2, y la copia 2 por B1, que ahora aprenderán que el nodo está en la interfaz superior.

Este tipo de situaciones no rompen la red, pero pueden generar problemas como que haya frames que permanecen viajando por la red sin parar.

Para solucionar esto, aparecen protocolos que **permiten** que se generen **ciclos físicos** (por sus beneficios en cuanto a redundancia), pero los **eliminan a nivel lógico**.

5.7. Spanning Tree Protocol

El Spanning Tree Protocol (STP) es un protocolo que construye una topología lógica sin ciclos para redes Ethernet (independientemente de si hay ciclos físicos).

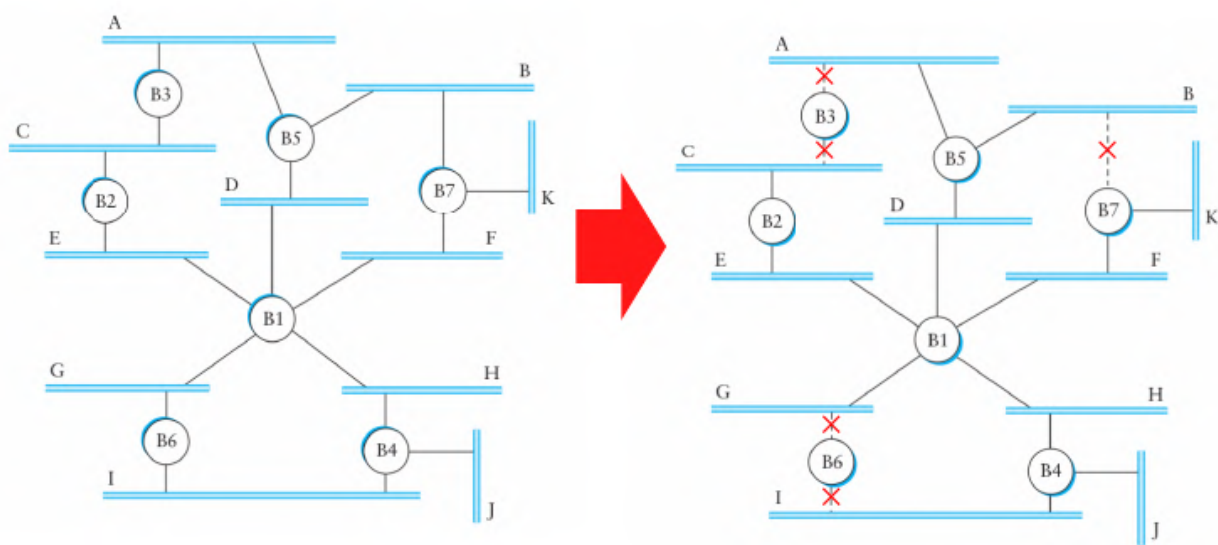


Figura 25: A la izquierda una red LAN con ciclos, y a la derecha la topología obtenida por ejecutar STP.

En la figura 25, los segmentos azules son los distintos segmentos de la LAN (que tendrían hosts conectados a ellos, aunque no estén dibujados), y los círculos son los bridges que conectan esos segmentos. La topología tiene tres ciclos. STP transforma ese grafo en uno acíclico, inhabilitando ciertas interfaces de los bridges para eliminar ese camino del grafo. Esto se mantiene actualizado y estable a lo largo del tiempo, sin necesitar un administrador.

5.7.1. Idea

Cada bridge/switch envía paquetes (llamados BPDUs, Bridge Protocol Data Units) a sus **vecinos**, propagando **información acerca de la topología** de la LAN de manera periódica.

Es decir, los switches mantienen una *foto*, el estado de un sistema distribuido, actualizándola periódicamente.

Cada switch decidirá si bloquea algunos de sus puertos con el objetivo de que la topología no tenga ciclos.

5.7.2. Mecanismo

- Se elige un switch como **root** (suele ser el de menor ID).
- Cada switch aprende las **distancias a root de todos sus vecinos**.
- Cada switch determina cuál es su **interfaz con distancia mínima** al root.
- Por cada LAN, se elige una única interfaz de un switch como **designada**, que tenga la distancia mínima a root entre las posibles.

5.7.3. Bridge Protocol Data Units

Los BPDUs están conformados por:

- El **ID del switch** que envía el mensaje.
- El **ID del root** según el switch que envía el mensaje.
- La **distancia** (en saltos) desde el switch que envía el mensaje hasta el root.

Esta información se **actualiza** en cada switch si:

- se identifica un BPDU con menor ID de root, o
- se identifica un BPDU con ID de root igual pero a menor distancia, o
- el ID de root y la distancia son las mismas pero el ID del switch es menor.

Los BPDUs deben llegar a todos los switches, por lo que su dirección de destino es la de **broadcast**.

La topología **converge** cuando a todos los switches les llega la misma información en los BPDUs (el estado es consistente en todos los nodos).

5.7.4. Estados de Interfaces

Las interfaces (los puertos de los switches) puede ser:

- **Root port**: el puerto con menor distancia al root, elegido de entre los puertos de cada switch.
- **Designated port**: todo puerto con mejor distancia al root, elegido de entre todos los puertos de varios switches conectados a una LAN.
- **Blocked port**: el resto de puertos.

5.8. LAN Virtual

Incluso habiendo segmentando la red en varias LANs, y con los learning bridges andando por haber usado STP, sigue siendo necesario hacer broadcasts y floodings (ya sea por los BPDUs, por tener falta de entradas en las tablas de reenvíos, etc.).

Esto se vuelve un problema cuanto más grande se hace la red: **el broadcast no escala**. Es decir, el dominio de broadcast se hace demasiado grande como para mantener una buena performance con una red en crecimiento.

Como consecuencia, las LANs extendidas no escalan. Un posible enfoque es crear **VLANs** (Virtual LANs). Esto es seguir la misma idea que para segmentar el dominio de colisión (con las LANs extendidas) pero con el dominio de broadcast: **particionar una LAN en varias LANs diferentes e interconectarlas a nivel lógico** (la topología física de la red no cambia; se usan clasificadores lógicos para las distintas LANs dentro de la VLAN). Esto hace que los broadcasts y las inundaciones sean más *localizadas*.

Sin embargo, esto no es tan automatizado: un administrador debe crear las VLANs (se hace una vez, vía software).

5.9. Problemas en Redes Inalámbricas

5.9.1. Problema de la Estación Oculta

Considerando un escenario como el que muestra la figura 26, cuando A transmite un mensaje a B: si C sensa el medio, no escuchará a A, pues está fuera de su alcance. Por lo tanto, *pensará* que puede transmitir.

Si C comienza a transmitir, generará una interferencia en B, eliminando la trama enviada por A.

Este problema, de que una estación no puede detectar a un potencial *competidor* por el medio, dado que éste se encuentra lejos, se denomina **problema de la estación oculta**.

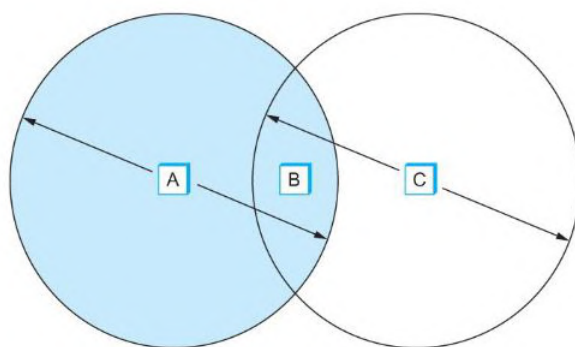


Figura 26: Esquema del problema de la estación oculta

5.9.2. Problema de la Estación Expuesta

Este es un problema que se presenta en un escenario similar al de la estación oculta. Observando la figura 27, pero considerando ahora la situación en la que B transmite un mensaje a A.

Si C sensa el medio, escuchará una transmisión, y *concluirá* que no puede enviar un mensaje a D. Sin embargo, una transmisión de C a D causaría una mala recepción sólo en la zona entre B y C, en la que no se encuentra localizado ninguno de los receptores (D).

Este problema, en el que una estación decide no transmitir un mensaje por sensar el medio ocupado, cuando en realidad la transmisión no generaría interferencias significativas, se denomina **problema de la estación expuesta**.

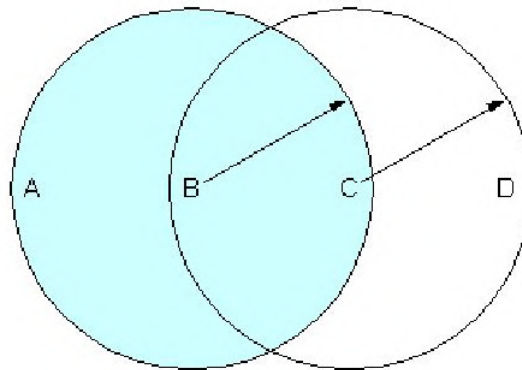


Figura 27: Esquema del problema de la estación expuesta

5.10. CSMA/CA

Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA) es un método de acceso múltiple en el cual se utiliza el sensado de la señal portadora (para permitir el acceso múltiple), pero los nodos intentan **evitar colisiones** empezando a transmitir únicamente después de que el medio se sence como *ocioso* (idle). Cuando los nodos transmiten, lo hacen con el paquete de datos entero.

Antes de transmitir, una estación debe determinar el estado del medio (libre u ocupado):

- Si el canal está **libre**, se realiza una espera adicional, llamada **espaciado entre tramas** (IFS, en Wi-Fi).
- Si el canal se encuentra **ocupado** o se ocupa durante la espera, se **espera** hasta el final de la transacción actual. Tras ello, se ejecuta el algoritmo de **backoff**, con el que se determina una **espera adicional y aleatoria**, elegida uniformemente en un intervalo llamado **ventana de contención** (CW). Esto se mide en *ranuras temporales* (slots). Si durante esta espera el medio no permanece libre durante un tiempo igual o superior al IFS, la espera queda suspendida hasta que se cumpla dicha condición.

A diferencia de lo que ocurre en Ethernet, cuando se transmite una trama en **Wi-Fi**, se **espera recibir un ACK**. Si no se recibe, se asume que ocurrió una colisión, y se **retransmite**.

El uso del backoff y las esperas aleatorias hace que la probabilidad de que ocurran colisiones sea baja, dado que es poco probable que dos nodos elijan el mismo factor de backoff.

5.11. Wi-Fi (IEEE 802.11)

Como se ve en la figura 28, en el nivel físico hay 4 alternativas de transmisión para las redes Wi-Fi:

- **Infrarrojo**: hoy en día muy poco usado por lento y poca distancia.
- **FHSS** (*Frequency Hopping Spread Spectrum*): sistema de bajo rendimiento, también muy poco usado.
- **DSSS** (*Direct Sequence Spread Spectrum*).
- **OFDM** (*Orthogonal Frequency Division Multiplexing*).

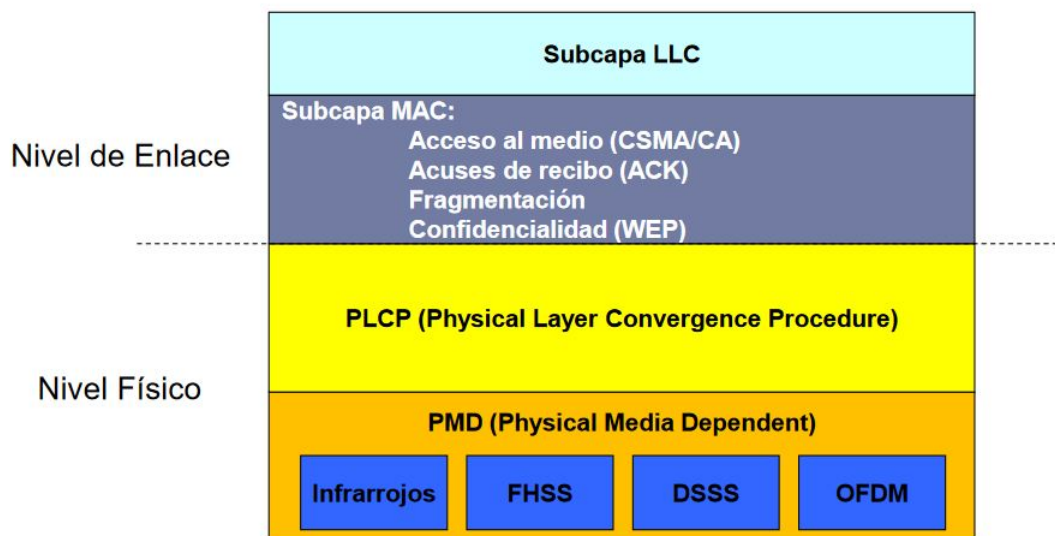


Figura 28: Modelo de referencia de 802.11.

Configuraciones típicas de la infraestructura:

- **BSS** (*Basic Service Set*): un *access point* provee la función de un puente (*bridge*) local para BSS. Todas las estaciones se comunican con el *access point* y no directamente entre ellas. Las tramas son retransmitidas entre las estaciones Wi-Fi por el *access point*.
- **ESS** (*Extended Service Set*): un ESS es un conjunto de BSSs, donde los access points se comunican entre ellos para forwardear el tráfico desde una BSS a otra.
- **AdHoc**: las estaciones inalámbricas se comunican directamente entre sí. Cada estación puede o no ser capaz de comunicarse con otra debido a las limitaciones de rango.

El medio utilizado por Wi-Fi (no guiado) es full-duplex (se puede enviar y recibir simultáneamente), pero implementar un protocolo full-duplex es muy costoso, así que el 802.11 se implementa como **half-duplex**. Por lo tanto, no es posible utilizar un mecanismo como CSMA/CD (detección de colisiones), pues un emisor no puede sensar efectivamente el medio mientras transmite un paquete (la potencia de la señal que transmite es muchísimo mayor que la recibida). Es por eso que aparece como alternativa CSMA/CA (prevención de colisiones) para las WLAN (Wireless Local Area Network).

Otro motivo por el que no es posible usar detección de colisiones en una WLAN es que no todas las estaciones de una red inalámbrica pueden *escucharse* entre sí, y que eso ocurra es una premisa para poder detectar colisiones.

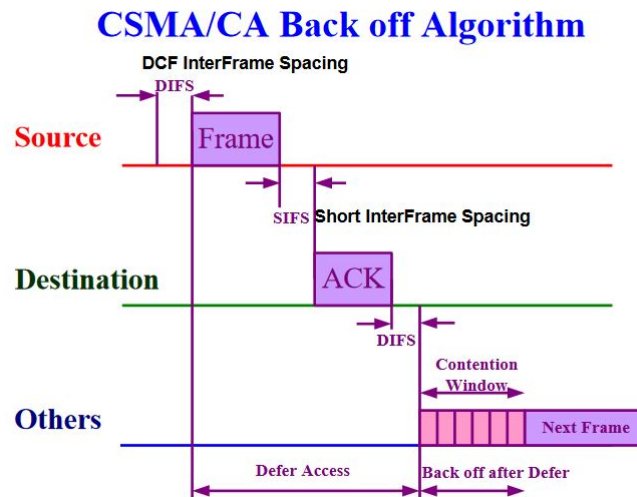


Figura 29: Algoritmo de CSMA/CA implementado en DCF.

Entonces, IEEE 802.11 MAC define dos métodos de acceso:

- **Distributed Coordination Function (DCF, *función de coordinación distribuida*)**: implementa CSMA/CA, lo cual, a priori, garantiza un acceso al medio equitativo (fairness). Este es el mecanismo base del protocolo.
 - Cuando un nodo sensa el canal y está libre, transmite.
 - Cuando un nodo sensa el canal y este se encuentra ocupado, **espera hasta que finalice la transmisión actual más un período de contención** (esto es lo que asegura el fairness). Este período de contención se cuantifica mediante un backoff counter: cuando un nodo recibe un frame para transmitir, elige un valor de backoff aleatorio, que determina cuánto tiempo deberá esperar hasta que pueda transmitirlo. Esto se almacena en el backoff counter, y mientras el canal está libre se decrementa su valor (si no, se mantiene). Cuando el backoff counter llega a 0, el nodo transmite el frame y espera un ACK. Si este ACK no llega, la **ventana de contención** se selecciona entre un intervalo aleatorio el **doble de grande**. Esto se repite hasta que el canal esté libre para transmitir.
- **Point Coordination Function (PCF, *función de coordinación de puntos*)**. Este es un mecanismo opcional.

5.11.1. Anomalía del Wi-Fi

En una red Wi-Fi, debido al mecanismo de acceso CSMA/CA (que asegura un acceso equitativo al medio compartido), se produce el fenómeno de que los nodos de baja velocidad degradan el throughput de los de alta velocidad. Los nodos reducen su data rate cuando la potencia de la señal es baja (Wi-Fi auto-rate).

Por lo tanto, los paquetes de los nodos de baja velocidad (por ejemplo, alguien en el baño de la casa, a 5 metros del access point, con un celular) consumen más *tiempo de aire* que los de alta velocidad (por ejemplo, otra persona al lado del access point, en una computadora): los **nodos lentos monopolizan el canal half-duplex**.

Además, Wi-Fi arbitra las transmisiones paquete a paquete, por lo que los nodos de velocidad alta reciben menos *tiempo de aire*.

5.11.2. MACA y MACAW

Dado que las técnicas de CSMA no funcionan bien en medios inalámbricos, se desarrollaron otras técnicas:

MACA (*Multiple Access Collision Avoidance*): se usó como base para el 802.11. El concepto en que se basa es que el transmisor estimula al receptor a enviar una trama corta, de manera que las estaciones cercanas puedan detectar esta transmisión y eviten ellas mismas de hacerlo durante la trama siguiente de datos.

Ejemplo: *A* comienza por enviar una trama RTS (*Request to Send*) a *B*. Esta trama corta (30 bytes) contiene la longitud de trama de datos que seguirá posteriormente. Entonces *B* contesta con una trama CTS (*Clear to send*). La trama contiene la longitud de los datos (copiado de la trama RTS). *A* la recepción de la trama CTS, *A* comienza a transmitir.

Cualquier estación que escuche el RTS está lo suficientemente cerca de *A* y debe permanecer en silencio durante el tiempo suficiente para que el CTS se transmita de regreso a *A* sin conflicto. Cualquier estación que escuche el CTS evidentemente está lo suficientemente cerca de *B* y debe permanecer en silencio durante el siguiente tiempo de transmisión de datos, cuya longitud puede determinar examinando la trama CTS.

Nuevamente esto no garantiza la ausencia de colisiones, pero reduce su probabilidad (por ejemplo *B* y *C* pueden intentar enviar una trama RTS al mismo tiempo).

MACAW: es una mejora de MACA que agrega los siguientes cambios:

- Agrega un ACK tras cada trama de datos exitosa.
- Agrega detección de portadora (CSMA/CA).
- Ejecuta el algoritmo de *exponential backoff* por separado para cada flujo de datos (no por estación).
- Agrega un mecanismo para que las estaciones intercambien información de congestión.

5.12. Fuentes

- Rodrigo Castro. Teoría de las Comunicaciones, Clase Teórica 3. Primer cuatrimestre, 2020.
- Claudio Righetti. Teoría de las Comunicaciones, Clase Teórica 10. Segundo cuatrimestre, 2021.
- Julián Sackmann. Teóricas de Teoría de las Comunicaciones. 2012.

6. Nivel de Red: Ruteo

6.1. Introducción

6.1.1. Sistemas Autónomos

Un sistema autónomo (AS, autonomous system) es un *pedacito de Internet*, es decir, en su interior tienen gran cantidad de enrutadores, pero tiene la particularidad de la autonomía y la heterogeneidad: los nodos que participan del sistema pueden ser distintos entre sí, pero hay cierto *acuerdo en común* que permite que funcionen en conjunto (por ejemplo, un algoritmo de ruteo consistente para un sistema).

Dos sistemas autónomos no tienen por qué utilizar los mismos algoritmos de ruteo, por lo que es necesario tener protocolos para que los sistemas se puedan comunicar entre sí, manteniendo la idea de que lo interno a un sistema autónomo es invisible para los demás.

Internet es una **interconexión de sistemas autónomos**.

6.1.2. Ruteo Interno y Externo

La idea será, continuando con el objetivo de la escalabilidad, poder armar **redes de redes**, donde cada nodo sea una de las redes como las de la sección anterior.

El **ruteo** se refiere a cómo llegar desde una red a otra. Armar esa ruta, y establecer una visión consistente de la misma son tareas que componen al ruteo, y son llevadas a cabo por los **routers**.

Los protocolos de ruteo se pueden clasificar en:

- **Ruteo interno** (IGP, Internal Gateway Protocols): su dominio de ruteos es dentro de un sistema autónomo (**intradominio**).
- **Ruteo externo** (EGP, External Gateway Protocols): aplican a interdominios, es decir entre distintos sistemas autónomos (**interdominio**).

6.1.3. Forwarding versus Routing

- El **forwarding** (reenvío) es un proceso para **seleccionar una puerta de salida** en base a la dirección de destino y las tablas de ruteo.
- El **routing** (ruteo) es el proceso mediante el cual son construidas y actualizadas las **tablas de ruteo**.

El ruteo es un problema de **grafos, optimización y algoritmos distribuidos**, pues la red se ve como un conjunto de nodos y arcos pesados; y el problema es encontrar el **camino de menor costo entre dos nodos**, en tiempo razonable y con recursos finitos.

(a)			
Destino Layer3	Prefix/Length	Next Hop	
	18/8	171.69.245.10	
(b)			
Destino Layer3	Prefix/Length	Interface	MAC Address
	18/8	if0	8:0:2b:e4:b:1:2

Figura 30: Ejemplo de entradas en las tablas de routing (a) y de forwarding (b). Estas dos tablas coexisten y están vinculadas, pero son construidas por protocolos distintos. La tabla de ruteo dice cuál es la dirección del siguiente *salto* para llegar al destino (dirección de capa 3), y la tabla de forwarding indica qué interfaz utilizar para dirigirse a la dirección de destino (de capa 2) que se corresponde con la dirección de capa 3 indicada por la tabla de ruteo.

6.1.4. Ruteo Estático y Dinámico

- **Estático:** configuración manual, no reactiva.
- **Dinámico:** configuración autónoma y adaptativa.

6.2. Protocolos de Ruteo Interno

Se verán dos protocolos de ruteo interno:

- **RIP**(Routing Information Protocol): el primero en usarse de manera extendida. Usa un algoritmo de *vector de distancias* (distance-vector), y se basa en la cuenta de *saltos* (hops).
- **OSPF**(Open Shortest Path First): más usado en la actualidad. Usa un algoritmo de *estado de enlaces* (link-state), y soporta balanceo de carga y QoS (Quality of Service, *Calidad de Servicio*), además de autenticación.

6.2.1. Distance-vector

Funcionamiento:

- Cada nodo mantiene una **tabla para todos los nodos**. Cada entrada de la tabla tiene:
 - Destino, identificación del nodo destino en el grafo.
 - Costo calculado hasta el momento, para llegar a ese destino.
 - Next hop (*próximo salto*), cuál es el siguiente nodo al que ir para llegar a ese destino (según lo calculado hasta el momento).

- Los nodos intercambian actualizaciones **sólo con sus vecinos**. Esto lo hacen tanto **periódicamente** (cada tantos segundos), **cuando su tabla cambia** o ante la **caída de un nodo** (actualización *gatillada*).
- Cada actualización es una lista de pares $\langle \text{Destino}, \text{Costo} \rangle$.
- Se modifica la tabla local de cada nodo **si se recibe una mejor ruta** que la calculada hasta el momento (es decir, de menor costo, o bien información proveniente de un next hop).
- Las rutas existentes se refrescan, y se borran si hay time-out (los nodos pueden desaparecer y volver a aparecer).
- El algoritmo usado para el cálculo de distancias es Bellman-Ford en versión distribuida. Notar que al tratarse de un algoritmo distribuido, ningún nodo tiene toda la información de la tabla, pero tiene una visión consistente de la red.

Estas actualizaciones distribuidas hacen que se converja a una visión consistente de la red. En caso de que uno de los enlaces deje de ser utilizable por algún problema, los nodos que estuvieran conectados a él detectan la falla; luego, deben fijar su distancia al vecino conectado vía ese enlace en infinito, y enviar actualizaciones a sus otros vecinos. Estos vecinos harán lo mismo si usaban al nodo en cuestión para alguna de sus rutas. Pero gracias a las actualizaciones periódicas, estos nodos podrán recibir la información de otro nodo vía sus vecinos, y reconfigurar sus rutas para no usar el enlace caído.

Problemas:

Por un lado la convergencia del protocolo no es buena, en comparación a otras alternativas como link-state.

Por otro lado, casos como el mencionado son los *felices*, pero pueden ocurrir situaciones que lleven a problemas, como pueden ser **inestabilidad** o **conteo a infinito**. Este último proviene de que si un nodo A le informa a otro nodo B que tiene un camino para ir hacia algún nodo, B no tiene forma de saber si ese camino tiene a B como uno de sus partes.

Ejemplo de conteo a infinito: suponer una red conectada en forma de grafo línea A-B-C-D-E-F. En algún momento, el nodo A se cae. Por lo tanto, a B no le llega la actualización periódica de A, por lo que su distancia a A, que era 1, ahora será infinita. Pero además, B recibe una actualización de C, que todavía no se había enterado de que A está caído. Entonces, C le informa a B que puede llegar a A en dos saltos ($C \rightarrow B \rightarrow A$). B no sabe que ese camino lo incluye a él, así que actualiza su tabla con distancia 3 para llegar a A. Luego, B envía la actualización a C, y como C es un next hop de B, C actualiza su tabla con distancia 4 para llegar a A. Esto se propaga a toda la red hasta que la distancia se convierte en infinito, por un ciclo de actualización.

Heurísticas para romper los ciclos:

- Fijar un número concreto como infinito (al llegar a un costo determinado, se asume que no hay ruta al nodo).
- Partir el horizonte (**split horizon**): cuando un nodo A recibe información de un nodo B y compute la nueva ruta, envía la información a todos sus vecinos **excepto a B**.
- Partir el horizonte con reverso venenoso (**split horizon with poison reverse**): cuando un nodo A detecta que una de sus rutas conectadas se cayó, va a “envenenar” la ruta

asignándole ∞ a su distancia y avisándolo a sus vecinos. Cuando uno de sus vecinos reciba esto, va a romper la regla de *split horizon* y anunciarle a todos sus vecinos (incluido A) de la ruta caída.

Las últimas dos técnicas sólo operan cuando el enlace involucra dos nodos. RIP utiliza una combinación de estas heurísticas.

Routing Information Protocol (RIP):

Este protocolo implementa distance-vector, utilizando como distancias válidas las menores a 16. Es decir, un costo de 16 representa infinito: esto es una clara limitación, dado que sólo servirá para **redes de tamaño pequeño**.

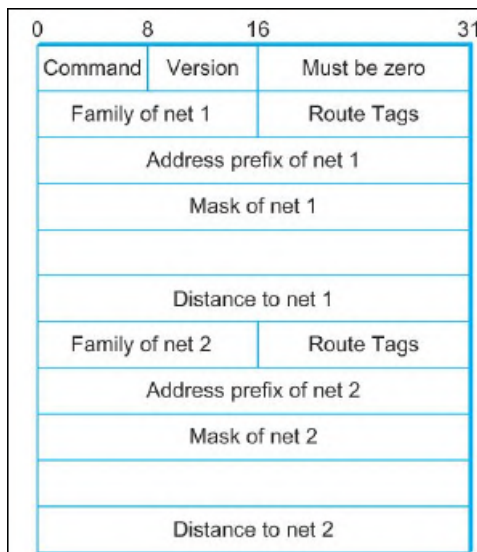


Figura 31: Formato del paquete RIPv2.

6.2.2. Link-state

En este caso, en vez de mandar toda la información de la red a los vecinos, se envía a toda la red información acerca de los vecinos.

Se asume que la topología de red y los costos son **conocidos por todos los nodos**, que tienen todos la misma información (link-state broadcast).

Funcionamiento:

- Cada nodo descubre a sus vecinos y conoce sus direcciones de red.
- Luego mide el costo para cada uno de sus vecinos.
- Se construye un paquete con todo lo que acaba de aprender.
- Se envía ese paquete a los demás nodos (routers).
- Calcula la ruta más corta a todos los nodos. Generalmente, esto se hace usando el algoritmo de Dijkstra (forward search).

La **estrategia** consiste en enviar a todos los nodos **información sobre los enlaces** directamente conectados. Esto se hace en forma de **inundación**.

Esta información se envía empaquetada en LSPs (Link State Packet), que contienen:

- El **identificador** del router que creó el LSP.
- El **costo** del enlace a cada vecino directamente conectado.
- El **número de secuencia** (SEQNO).
- El **time-to-live** (TTL) para el LSP.

Este tipo de envío de información se conoce como **inundación confiable**, y consiste en que cada router:

- **Almacena** el LSP más reciente de cada nodo.
- **Decrementa** el TTL de cada LSP almacenado, descartándolos cuando TTL llega a 0.
- **Reenvía** el LSP a todos los nodos, **excepto a quien se lo envió**.
- **Genera** un nuevo LSP periódicamente, incrementando SEQNO.
- **Inicia** SEQNO en 0 cuando se reinicia.

Se dice que es *confiable* porque existen los paquetes de acknowledgement (**LSP ACK**) para proveer confiabilidad al envío y recepción de mensajes.

Los LSPs se generan **periódicamente** (en el orden de horas), o ante un **cambio en la topología**.

Concretamente:

- Los nodos almacenan los LSP que reciben en sus tablas.
- El pasaje de LSP entre vecinos se asegura mediante mecanismos de ACK y retransmisión.
- Si un nodo recibe un LSP de X y no lo tenía almacenado, lo almacena y lo propaga.
- Si un nodo recibe un LSP de X y sí lo tenía almacenado, verifica el número de secuencia:
 - si el recién llegado es más nuevo (mayor SEQNO) que el almacenado, se lo queda y lo retransmite a todos los nodos, menos al que se lo envió.
 - si el recién llegado es más viejo o igual que el almacenado, lo descarta.
- El hecho de no volver a enviar el paquete de vuelta al nodo del que se lo recibió ayuda a que la inundación termine.
- Como los nodos envían la información a todos sus vecinos conectados, entonces la información más reciente eventualmente alcanzará a todos los nodos.

Ventajas: Respecto de distance-vector, los protocolos basados en link-state son más estables, no generan tanto tráfico y responden rápidamente a cambios de topología.

Problemas: El principal problema es la cantidad de información que debe ser almacenada en los nodos (un LSP por nodo), lo cual es exigente en cuanto a memoria. Además, al ser un cómputo centralizado para el algoritmo de camino mínimo, requiere de más poder de procesamiento.

Open Shortest Path First (OSPF):

- Utiliza un algoritmo de **link-state** (distribución de paquetes del estado de los enlaces, mapa de la topología en cada nodo, cómputo de la ruta usando el algoritmo de Dijkstra).
- Tipos de mensajes:
 - Hello: descubre quiénes son los vecinos.
 - Link state update: proporciona los costos del emisor a sus vecinos.
 - Link state ACK: confirma la recepción de la actualización del estado del enlace.
 - Database description: anuncia qué actualizaciones tiene el emisor.
 - Link state request: solicita información del socio.
- Los anuncios se distribuyen a todos los nodos dentro de un sistema autónomo, mediante **inundación**.
- **Seguridad**: todos los mensajes de OSPF están autenticados.
- **OSPF jerárquico**: se dividen los sistemas autónomos en áreas: área local y área troncal. Los anuncios de link-state se hacen por área. Por lo tanto, aparecen cuatro tipos de routers: de frontera (conectan con otros AS), troncales, de frontera de área (dentro de un AS, separando las áreas), e internos (dentro de cada área). Esta jerarquización implica ocultar información, a costa de afectar la optimalidad de las decisiones (porque no todos los routers conocen a todos los del sistema autónomo). Es un clásico ejemplo del **trade-off entre escalabilidad y optimalidad**.

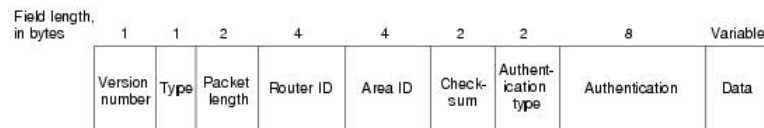


Figura 32: Formato del paquete OSPF.

6.2.3. Distance-vector vs Link-state

Distance Vector	Link State
Más liviano (no hay algoritmos de grafos complejos)	Más pesado (calcular Dijkstra en cada nodo)
Nodos bobos y baratos	Se necesita mucho almacenamiento y poder de cómputo en los nodos
Cada nodo transmite a sus vecinos lo que sabe respecto de toda la red (distancia a todos los nodos)	Cada nodo transmite a toda la red lo que sabe de sus enlaces vecinos (el estado de sus vecinos)
Más propenso a errores	Más estable
Mucho overhead	No genera tanto tráfico
Si se me cae un nodo puede entrar en el problema de conteo a infinito	Responde más rápido a cambios de topología
Escala peor	Escala mejor

Cada Router	DISTANCE-VECTOR	LINK-STATE
Qué informa ?	• Toda su Tabla de Ruteo	• Solo el Estado de sus Enlaces directos
A quién le pasa información ?	• Solo a sus vecinos	• A toda la red (inundación)
Algoritmo utilizado	• Bellman-Ford Distribuido	• Dijkstra
Datos utilizados	• Información de los vecinos	• Estado de Enlaces de cada nodo
Estructuras de Datos	• Tabla de Distancias • Tabla de Ruteo	• Tabla de Estado de Enlaces • Tabla de Ruteo
Características	• Ciclos de Ruteo Gran variedad de Algoritmos: • Merlin-Segall • Jaffe-Moss • Esquema OP • Diffusing Comp • Cheng • Cálculo Distribuido	• Visión Consistente de la Red • Gran uso de CPU y Memoria • Algoritmo Básico único • Cálculo Centralizado
Ejemplo de Protocolos de Internet	RIP	OSPF

Figura 33: Comparación entre dos algoritmos de ruteo interno: distance-vector y link-state, implementados por RIP y OSPF respectivamente.

6.3. Protocolos de Ruteo Externo

Tanto RIP como OSPF son protocolos de ruteo interno, es decir que funcionan dentro de un sistema autónomo. Independientemente de qué protocolo usen dos sistemas autónomos, deben poder comunicarse entre sí. Para esto están los protocolos de ruteo externo (ruteo interdominio).

Estos son los EGP (Exterior Gateway Protocols):

- Generalidades:
 - Están diseñados para una red de redes estructurada como un árbol.
 - Se preocupan por alcanzar los nodos, no por optimizar rutas.
- Mensajes del protocolo:
 - Adquisición de vecinos: los routers son pares entre ellos, e intercambian información de enlace.
 - Alcance de vecinos: un router periódicamente prueba si otro es alcanzable, con mensajes HELLO/ACK.
- Actualización de rutas:
 - Los nodos pares intercambian periódicamente sus tablas de ruteo (vector de distancias).

6.3.1. Border Gateway Protocol (BGP)

- Es un EGP diseñado para intercambiar información de ruteo y alcanzabilidad entre sistemas autónomos en Internet (red de redes).
- Cada sistema autónomo está bajo el control de una única entidad administrativa.
- Es otra forma más de agregar jerárquicamente información de enrutamiento para alcanzar gran escala.
- Se ocupa del ruteo afuera de los sistemas autónomos, mientras que cada sistema autónomo puede ejecutar su propio ruteo intradominio.
- Evita que diferentes sistemas autónomos compartan información de alcanzabilidad entre ellos: dispone de la información de qué rangos de direcciones IP se puede alcanzar en cada sistema autónomo, además de las rutas para llegar de uno a otro.

6.3.2. BGP y los Network Access Points

Los NAPs (Network Access Points), también conocidos como IXs (Internet eXchanges) son componentes fundamentales de la red de Internet. Son puntos neurálgicos de la red, y se han construido en todo el mundo bajo distintos esquemas institucionales, topológicos y operacionales.

A través de un NAP se produce el intercambio de tráfico entre las redes de diversas entidades.

La mayoría de los NAPs persigue objetivos comunes: hacer eficiente el ruteo en Internet, mejorando la calidad de servicio y minimizando los costos de interconexión.

6.4. Fuentes

- Rodrigo Castro. Teoría de las Comunicaciones, Clase Teórica 4. Primer cuatrimestre, 2020.
- Julián Len. Teoría de las Comunicaciones, Clase Práctica 6. Primer cuatrimestre, 2020.
- Julián Sackmann. Teóricas de Teoría de las Comunicaciones. 2012.

7. Nivel de Red: IP

7.1. Internetworking

Problemas a tratar cuando se conectan redes:

- Heterogeneidad: los usuarios de un tipo de red quieren comunicarse con usuarios de otros tipos de redes. Conectar hosts de dos redes diferentes puede requerir pasar por redes intermedias, cada una con su tecnología subyacente.
- Escalabilidad: por un lado, hay que encontrar un camino eficiente en una red de millones de nodos (ruteo), y por otro lado, dar identificadores para todos esos nodos (direccionamiento).

El objetivo es construir redes escalables, que puedan llegar a tener un tamaño considerable, con protocolos que permitan automatizar la mayor parte de las tareas involucradas en la transmisión y recepción de un paquete. Esto implica conectar redes entre sí, para lo cual se pueden usar switches.

Un **switch** es un dispositivo que **interconecta enlaces** para formar redes más grandes. Su trabajo es lograr que la mayor cantidad de paquetes que entren, vayan a la salida apropiada.

- Envía paquetes de un puerto de entrada a un puerto de salida (**switching, forwarding**).
- El puerto de salida se selecciona usando una dirección que trae el encabezado del paquete.

Para distribuir los paquetes, hay switches que usan **circuitos virtuales** y otros **conmutación de paquetes**, los cuales pueden ser de longitud fija o variable.

Al interconectarse entre ellos, los switches permiten cubrir mayores áreas geográficas, construyendo redes más grandes y ofreciendo tolerancia a la latencia (buffers).

7.2. Conmutación de Paquetes

Hay dos grandes paradigmas para la conmutación de paquetes:

1. Sin conexión (datagramas): la información se subdivide en paquetes (datagramas, unidades de información), y hay distintos caminos posibles para llegar desde el origen al destino. Cada paquete que compone un mensaje puede tomar un camino diferente en la red, pudiendo por ejemplo llegar fuera de orden.
2. Orientado a conexión (circuito virtual): en este caso sí se establece una conexión antes de empezar la comunicación; todos los paquetes que conforman el mensaje siguen el mismo camino desde el origen hasta el destino.

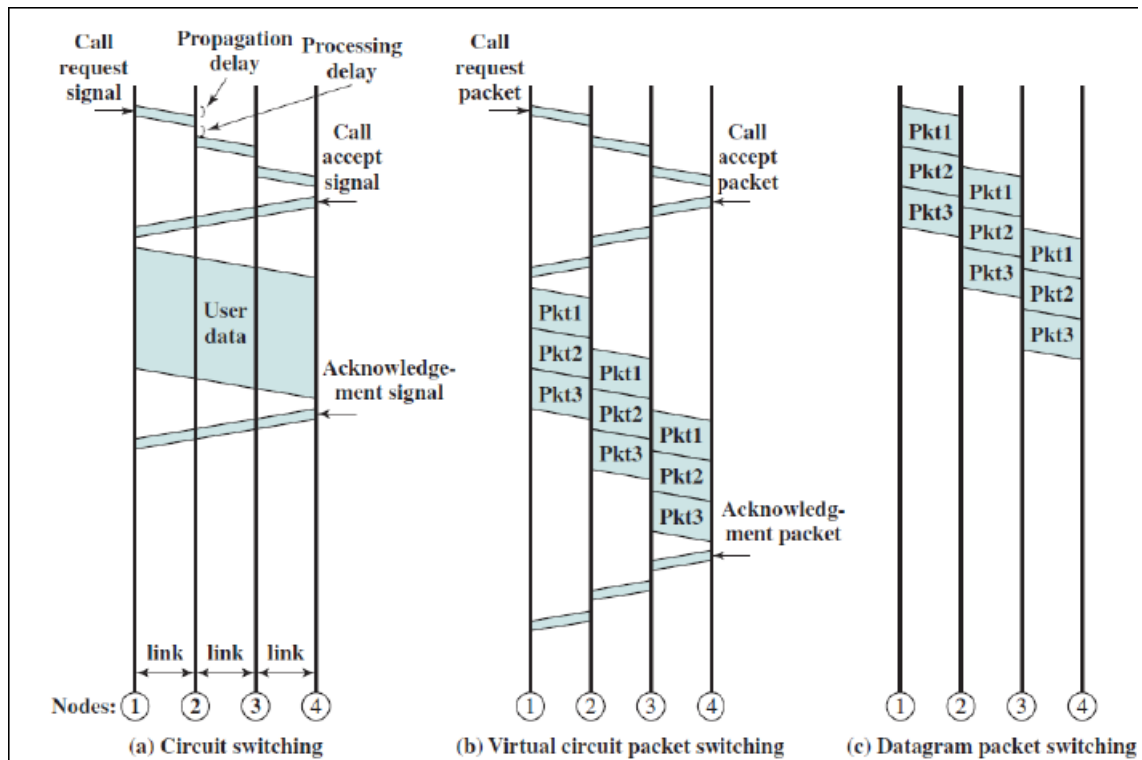


Figura 34: Temporización de eventos en los distintos paradigmas de comunicación.

7.3. Conmutación sin Conexión (datagramas)

- No hay una fase previa a la comunicación para establecer una conexión.
- El nodo puede enviar el paquete *cuando quiera*.
- Cada paquete se envía independientemente, por lo que debe llevar toda la información necesaria para alcanzar su destino.
- Analogía con el **sistema postal**.
- No hay que esperar un RTT para establecer una conexión: un nodo puede enviar los datos **cuando esté listo** (es más rápido que un modelo orientado a conexión).
- El nodo de origen no necesita saber si la red es capaz de entregar un paquete, o si el destino está listo para recibirlo.
- Es posible ir cambiando el camino que siguen los paquetes, porque estos son **tratados independientemente**. Esto sirve para evitar enlaces y nodos que estén fallando.
- La **información adicional de control** (overhead) que llevan los paquetes es mucho mayor que en un modelo orientado a conexión.

7.4. Conmutación Orientada a Conexión (circuitos virtuales)

- Se requiere una **fase inicial** para establecer la conexión, y una **fase de finalización** de la conexión, durante las cuales no se transporta datos del usuario (goodput nulo).
- Esto implica que hay que esperar un RTT completo antes de empezar a enviar información.

- Los paquetes que se transmiten **siempre usan el mismo circuito**.
- Analogía con la **llamada telefónica**.
- Cada switch mantiene una **tabla de VC** (virtual circuits), que contiene:
 - El puerto por el cual llega el paquete.
 - El identificador del circuito virtual (VCI) de entrada.
 - El puerto por el cual debe salir el paquete.
 - El identificador del circuito virtual (VCI) de salida.
- Al almacenarse la información sobre el camino en las tablas de circuitos virtuales de los switches, y ser un camino inmutable, la información de control que deben llevar los paquetes es mucho menor que en un modelo de datagramas.
- La solicitud de conexión debe llevar la dirección completa del nodo destino, pero los paquetes sólo llevan el VCI (overhead de transmisión pequeño).
- Si un switch o enlace falla, el circuito falla y se debe restablecer la conexión (overhead de recuperación ante errores grande).

7.4.1. Tipos de Conexiones

1. Conexión Permanente (PVC):

- Definida y finalizada por el administrador de la red.
- Un usuario solicita a la red la creación de registros en las tablas de VC.
- Después de creado el circuito virtual, se puede enviar datos.
- Para finalizar la conexión, el administrador de la red se encarga de *bajar* el circuito virtual.
- El circuito creado permanece fijo para ser reutilizado más adelante.

2. Conexión por Solicitud (SVC):

- Cuando el nodo A quiere enviar datos al nodo B, envía un mensaje de solicitud de conexión a la red.
- El switch que la recibe la envía al siguiente, y así hasta llegar a B.
- Si B acepta la conexión, devolverá un identificador de circuito se va a utilizar.
- Esta aceptación se repite en todos los switches del camino, y luego se empieza a enviar los datos.
- Cuando A no desea enviar más datos a B, termina el circuito virtual enviando un mensaje de finalización a la red.
- El switch que recibe el mensaje borra la entrada de la tabla de VC correspondiente a ese circuito, y eso se repite para cada switch hasta alcanzar a B.
- Una vez que A cierra la conexión, el circuito se desmonta, por lo que si lo quiere volver a usar más adelante, habrá que volver a repetir el proceso.

7.5. Datagrama vs Circuito Virtual

Asunto	Subred de datagramas	Subred de circuitos virtuales
Configuración del circuito	No necesaria	Requerida
Direccionamiento	Cada paquete contiene la dirección de origen y de destino	Cada paquete contiene un número de CV corto
Información de estado	Los enrutadores no contienen información de estado de las conexiones	Cada CV requiere espacio de tabla del enrutador por conexión
Enrutamiento	Cada paquete se enruta de manera independiente	Ruta escogida cuando se establece el CV; todos los paquetes siguen esta ruta
Efecto de fallas del enrutador	Ninguno, excepto para paquetes perdidos durante una caída	Terminan todos los CVs que pasan a través del enrutador
Calidad del servicio	Difícil	Fácil si se pueden asignar suficientes recursos por adelantado para cada CV
Control de congestión	Difícil	Fácil si pueden asignarse por adelantado suficientes recursos a cada CV

Figura 35: Comparación entre los dos paradigmas de conmutación de paquetes.

Datagramas	Conexiones
Los paquetes van por donde quieren	Los paquetes de la conexión van todos por el mismo camino
Comienzo inmediato	Esperar 1 RTT para comenzar a transmitir
Paquetes pesados	Paquetes livianos
Paquetes llegan desordenados	Paquetes llegan ordenados
Se banca que se caiga algún enlace/nodo	Si se cae un nodo de la conexión, se cae la conexión
No se permite reservar recursos	Permite reservar recursos en los switches

7.6. Internet Protocol (IP)

Recuerdo: la idea es interconectar redes a través de routers para lograr mayor escalabilidad. Cada una de las redes interconectadas puede tener una tecnología distinta (Ethernet, Wi-Fi, FDDI, Point-to-Point, etc.).

7.6.1. Modelo de Servicio IP

- **Sin conexión** (basado en datagramas).
- **Best-effort** (*mejor esfuerzo*). Esto implica que no hay muchas garantías en cuanto a la transmisión de mensajes: los paquetes pueden perderse, llegar fuera de orden, tener duplicados, y no tener una cota para el tiempo de entrega.

- Para proveer mayores garantías, se utiliza la capa superior (transporte).

7.6.2. El Encabezado IPv4

El **encabezado** (header) de un datagrama IP contiene información que deberá ser interpretada por los routers.

El **tamaño** de un encabezado es normalmente de 20 bytes, pudiendo llegar a 60 bytes si se usan todos los campos opcionales.

Después del encabezado vienen los datos como tales.

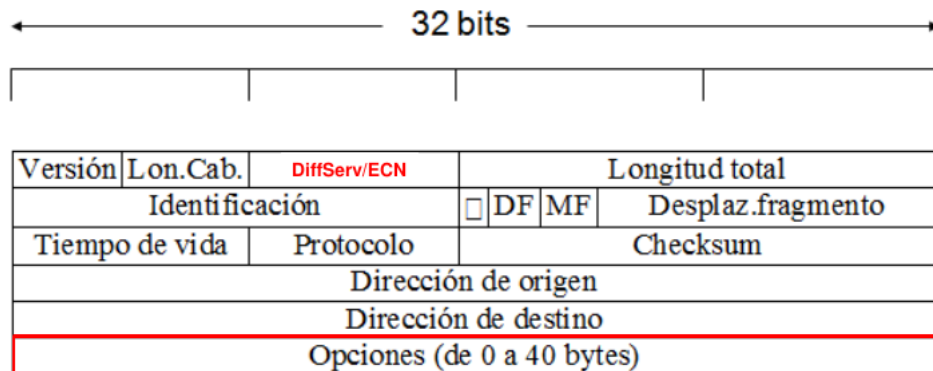


Figura 36: Formato de un encabezado de un datagrama IP.

En la figura 36 se pueden ver los campos de un header de IP:

- Versión: normalmente v4, aunque ya se usa v6 también (el formato del header varía según la versión).
- Longitud de encabezado: expresada en palabras de 32 bits.
- Longitud total: expresada en bytes, con máximo de 65.535 (incluye el header).
- Fragmentación: identificación, DF, MF, desplazamiento.
- Tiempo de vida (TTL): contador de saltos hacia atrás (se descarta el paquete cuando llega a 0).
- Protocolo: es el protocolo al que corresponden los datos (puede ser ICMP, IGMP, IP, TCP, UDP, OSPF, entre otros...).
- Checksum: sólo para la cabecera, no incluye los datos.
- Dirección fuente y destino: ambas de 32 bits.
- DiffServ/ECN: servicios diferenciados (calidad de servicio) / control explícito de congestión.

7.6.3. Fragmentación

A **nivel de enlace** (capa inferior), cada tecnología de red tiene un **MTU** (Maximum Transmission Unit, *unidad máxima de transmisión*), que varía según la tecnología.

Entonces, IP se debe **adaptar** a la tecnología de red subyacente (esas tecnologías no conocen a IP). Es decir, IP debe lidiar con el hecho de no poder enviarle un paquete de cualquier tamaño a la capa inferior.

Fragmentación: ocurre si un router recibe un datagrama que debe reenviar a una red en la que el MTU es menor que el tamaño de dicho datagrama. Es decir, *corta el paquete en pedacitos*.

Reensamblado: del otro lado, en el host de destino, se debe reensamblar los pedacitos del datagrama recibido (no lo hacen los routers intermedios, sino el host final).

Todos los fragmentos tienen el **mismo identificador**, y son datagramas autocontenidos, con su header correspondiente, que es igual que el del datagrama original (salvo por los campos de fragmentos y desplazamiento: MF, Más Fragmentos, y DF, Desplazamiento del Fragmento).

En tanto modelo best-effort, IP no recupera los fragmentos perdidos.

7.6.4. Direccionamiento Global

Cada host y router en Internet tiene al menos una **dirección IP**: hay una dirección por cada **interfaz**.

Propiedades de la dirección IP:

- Dirección globalmente única.
- Jerárquica: red + host.
- Largo de 32 bits.
- Notación dot: cuatro números entre 0 y 255, separados por puntos (e.g. 192.12.69.77, 128.96.33.81).
- Esquema de clases (classful): clases A, B, C.
- Problemas de escalabilidad.

Class	Leading bits	Size of <i>network number</i> bit field	Size of <i>rest</i> bit field	Number of networks	Addresses per network	Start address	End address
A	0	8	24	128 (2^7)	16,777,216 (2^{24})	0.0.0.0	127.255.255.255
B	10	16	16	16,384 (2^{14})	65,536 (2^{16})	128.0.0.0	191.255.255.255
C	110	24	8	2,097,152 (2^{21})	256 (2^8)	192.0.0.0	223.255.255.255

Figura 37: Las tres clases de direcciones IP y sus características.

Los hosts y los routers interpretan las direcciones IP separándolas en **red** y **host**. En el sistema classful (1981), los largos de estas dos partes son fijos, pero esto tiene problemas para escalar. Actualmente se utiliza otro sistema con subredes (1985) y sin clases (CIDR, 1993).

7.6.5. Direcciones IP Classful

En la figura 38 se pueden ver las tres clases básicas de direcciones IP:

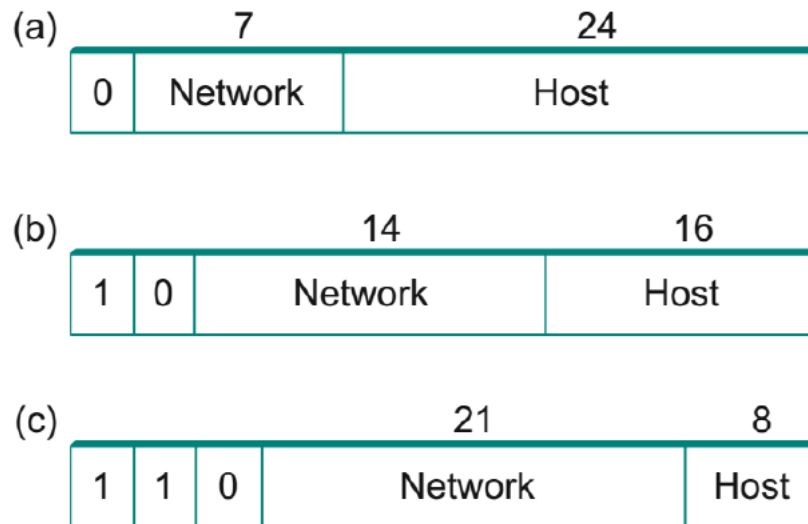


Figura 38: Formato para cada clase de dirección

- Clase A: más direccionamiento para hosts, menos para redes. Pensado para pocas redes pero con muchos hosts.
- Clase B: más balanceado entre direccionamiento para redes y hosts.
- Clase C: mucho direccionamiento para redes, poco para hosts. Dirigido a la gran mayoría de redes, que no tienen tantos hosts conectados.

Los principales problemas de este modelo son:

- Falta de flexibilidad interna.
- Uso ineficiente del espacio de direcciones.
- Crecimiento de las tablas en routers.

7.6.6. Direcciones y Máscaras

La longitud de cada parte de la dirección IP (red y host) se indica mediante la **máscara de red**. Ésta tiene una longitud de 32 bits, y está formada por un conjunto de unos seguido de ceros. **Los unos indican la parte de red**.

Al igual que las direcciones, las máscaras se expresan mediante cuatro números separados por puntos (e.g. 255.255.255.0 dedica 28 bits a direccionar redes, y 4 bits a los hosts).

También existe la notación concisa, que codifica con una barra seguida de un número que representa la cantidad de unos de la máscara (por ejemplo, la máscara 255.224.0.0 sería /11).

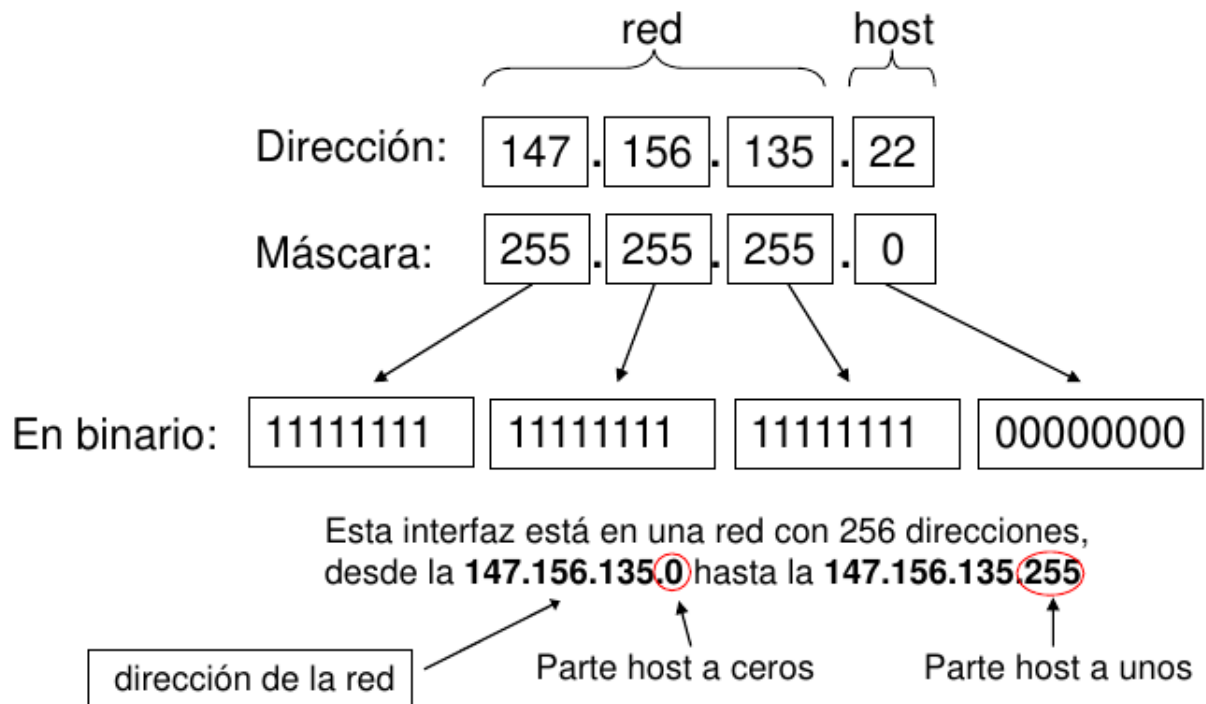


Figura 39: Ejemplo de dirección IP y máscara.

7.6.7. Direcciones Especiales

Para cada red (e.g. 40.40.25.0 con máscara 255.255.255.0):

- La **primera dirección** posible, es decir dejando la parte de host en cero, **identifica la red** (e.g. 40.40.25.0).
- La **última dirección** posible es la de **broadcast** en esa red (e.g. 40.40.25.255).
- El rango asignable es el que queda en medio de estas dos direcciones reservadas (e.g. desde 40.40.25.1 hasta 40.40.25.254).

Dirección	Significado	Ejemplo
255.255.255.255	Broadcast en la LAN propia	255.255.255.255
0.0.0.0	Depende: "Este host", "cualquier IP en este host", "cualquier IP", "default route"	0.0.0.0
Parte Host a ceros	Identifica una red	147.156.0.0 255.255.0.0
Parte Host a unos	Broadcast en una red remota	147.156.255.255 255.255.0.0
127.0.0.1	Dirección Loopback (para pruebas)	127.0.0.1

Figura 40: Direcciones especiales y reservadas

7.6.8. Asignación de Direcciones IP

La asignación de direcciones y máscaras se puede hacer:

- **Manual:** configurándolo a mano en el propio equipo.
- **Automáticamente:** mediante un protocolo de asignación desde un servidor (e.g. DHCP).

Además, es común asignarle al host un router por defecto (**default gateway**, o *puerta de acceso predeterminada*).

7.6.9. Direcciones IP Privadas

Existen tres rangos de direcciones IP declarados como **privados**:

1. 10.0.0.0 - 10.255.255.255/8
2. 172.16.0.0 - 172.31.255.255/12
3. 192.168.0.0 - 192.168.255.255/16

Estas direcciones se pueden usar internamente en organizaciones como sea conveniente, mientras se cumpla que los paquetes que las contengan no aparezcan en Internet.

Son útiles para:

- Dispositivos que no requieren conexión a Internet (e.g. impresoras, switches...).
- Interconectar sensores con las computadoras que los controlan.
- Redes de usuarios o servidores que no deban acceder a Internet.
- Redes que accedan a Internet mediante otros mecanismos (e.g. NAT-PAT, proxy...).
- Paliar la escasez de direcciones IP públicas disponibles.

7.6.10. Forwarding

- Cada datagrama tiene la **dirección de destino**.
- Cuando un router recibe un datagrama:
 - Si **está directamente conectado** a la red de destino, hace un **forward al host**.
 - Si **no está directamente conectado** a la red de destino, hace un **forward a otro router**.
- Tabla de forwarding: conecta un número de red (cada una identificada por una dirección unívoca) al next hop (el *próximo salto*).
- Cada host tiene un **default router**, al cual recurre para *salir al mundo*.
- Cada router mantiene una tabla de forwarding.

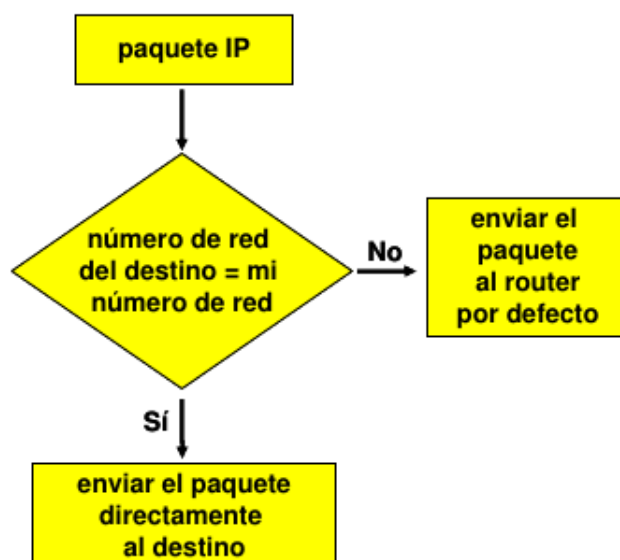


Figura 41: Esquema del algoritmo de forwarding para un host.

Es posible que haya varias rutas válidas para un mismo paquete (por ejemplo, la ruta por defecto aplica para cualquier paquete). En estos casos, se revisa primero las rutas de máscara más larga.

Esto garantiza que se aplicarán **primero las rutas más específicas**, y luego las más generales.

Por ejemplo, si un router recibe un datagrama con destino 200.40.1.1, y la búsqueda en la tabla encuentra dos entradas: 200.40.1.0/24 y 200.40.0.0/16, la ruta elegida será la primera.

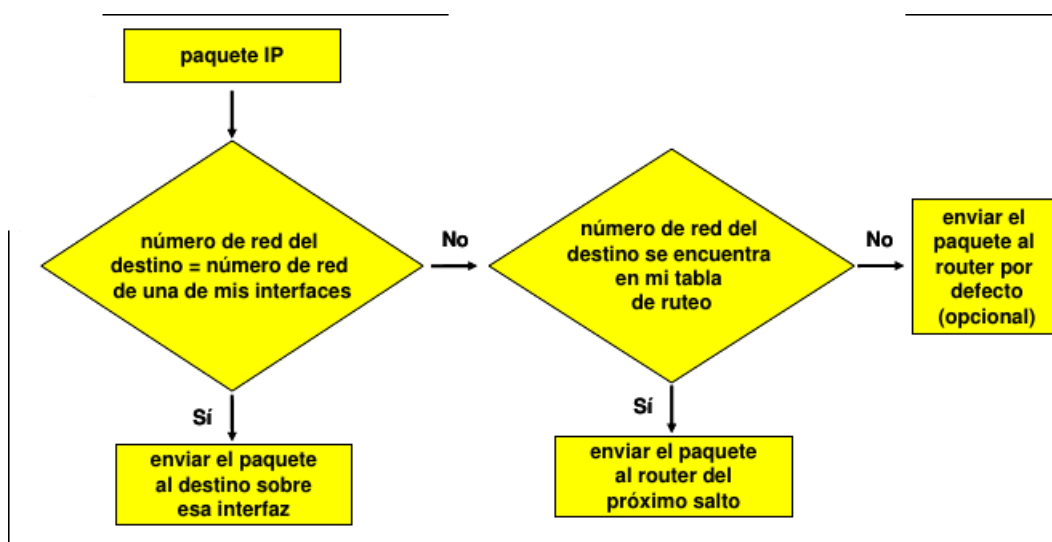


Figura 42: Esquema del algoritmo de forwarding para un router.

7.6.11. Subredes IP

Frecuentemente, la red de una organización se compone de otras redes, por lo que es conveniente dividir una red en trozos más pequeños llamados subredes.

Por ejemplo, si una empresa utiliza la red 40.40.0.0 con máscara 255.255.0.0 (desde 40.40.0.0 hasta 40.40.255.255), para reducir el tráfico broadcast se puede dividir la red en subredes, todas

con 256 o menos hosts. En la figura 43 se puede ver una posible configuración:

VLAN	Subred	Máscara	Rango
1	40.40.0.0	255.255.255.0	40.40.0.0 - 40.40.0.255
2	40.40.1.0	255.255.255.0	40.40.1.0 - 40.40.1.255
3	40.40.2.0	255.255.255.0	40.40.2.0 - 40.40.2.255
...
256	40.40.255.0	255.255.255.0	40.40.255.0 - 40.40.255.255

Figura 43: Subredes para el ejemplo.

Las subredes son útiles para:

- Reducir el tráfico broadcast en una red local grande.
- Conectar redes locales remotas de una organización usando routers y enlaces punto-a-punto.
- Motivos de seguridad: separar redes a nivel IP y usar técnicas de filtrado en el router.
- Dividir una red local en zonas con distinto nivel de seguridad (DMZs en un firewall).
- Separar las redes de servicios, clientes, backbone, etc. en un ISP.

7.6.12. Máscaras de Tamaño Variable (VLSM)

A veces conviene dividir una red en **subredes de diferentes tamaños**. Para eso, se pueden usar **máscaras de tamaño variable**: la división red/host no es igual en todas las subredes.

Aunque las subredes pueden tener diferente tamaño, **no pueden solaparse** pues de hacerlo existirían direcciones duplicadas.

7.6.13. Direcciones sin Clases (CIDR)

En el Classless Inter-Domain Routing (CIDR, 1993), se deja de utilizar el sistema de clases A, B y C, y los bloques de IP se pueden asignar con cualquier prefijo.

También introduce la *agregación de rutas* (route aggregation), lo cual reduce el tamaño de las tablas en routers.

Su introducción tuvo como objetivo desacelerar el crecimiento de las tablas de ruteo en Internet, y el agotamiento de las direcciones IPv4 disponibles.

7.7. ARP

El Address Resolution Protocol (ARP) es un protocolo usado para obtener la dirección de nivel de enlace (MAC) asociada a una dirección de nivel de red (IP). Por eso se dice que es un protocolo de capa 2.5.

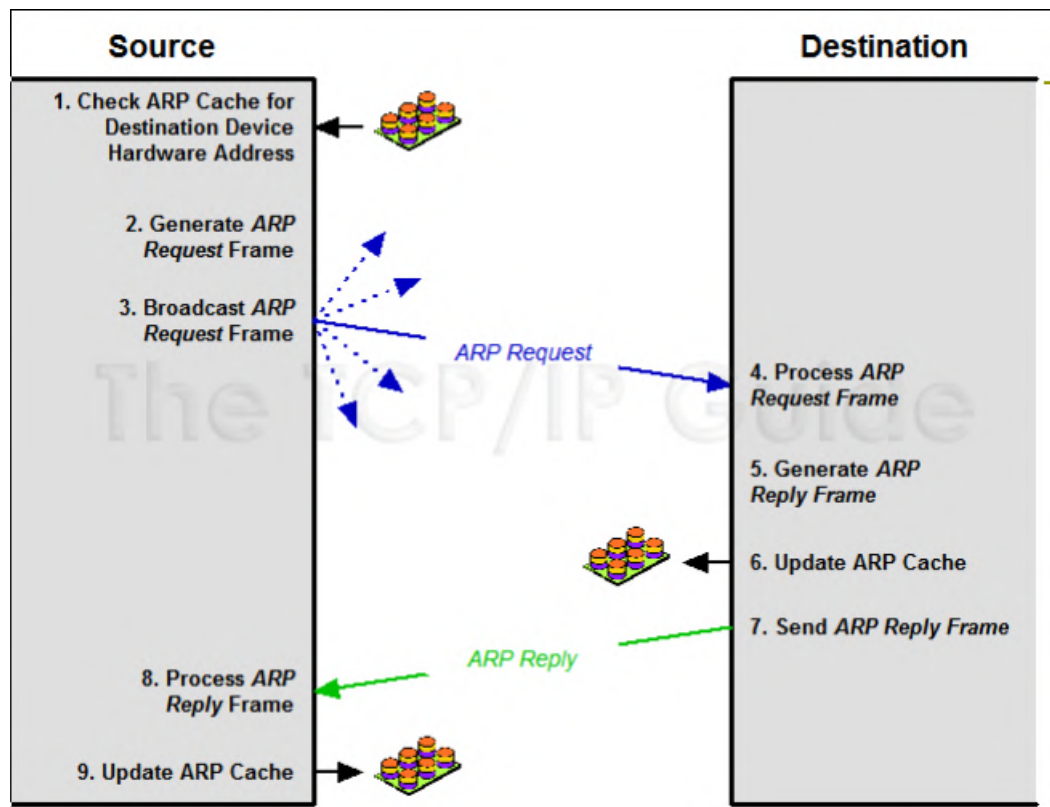


Figura 44: Proceso de transacción ARP.

7.8. ICMP

El Internet Control Message Protocol (ICMP) es un protocolo auxiliar de control para la suite de IP. Es usado por dispositivos de nivel de red (e.g. routers) para enviar mensajes de error e información de control indicando éxito o fallo al comunicarse con otra dirección IP.

Permite reportar incidencias o situaciones excepcionales que se pueden producir en el envío de un datagrama.

Todos los mensajes ICMP se envían en datagramas IP, y pueden ser de error (e.g. host unreachable, reassembly failed, TTL reached 0, checksum failed...) o de feedback (e.g. redirect, echo, timestamp...).

También posibilita implementar herramientas de diagnóstico para problemas de red, como **ping** (usando echo request y echo reply para comprobar la accesibilidad de una IP) y **traceroute** (usando time exceeded para detectar cuando un datagrama es descartado por agotamiento del TTL, lo cual se usa para determinar la ruta efectuada por un paquete hacia un destino en la red).

7.9. Fuentes

- Rodrigo Castro. Teoría de las Comunicaciones, Clase Teórica 5. Primer cuatrimestre, 2020.
- Leonardo Balbiani. Teoría de las Comunicaciones, Clase Práctica 4. Primer cuatrimestre, 2020.
- Marcos Cervetto. Teoría de las Comunicaciones, Clase Práctica 5. Primer cuatrimestre, 2020.
- Julián Sackmann. Teóricas de Teoría de las Comunicaciones. 2012.